



# Sun Fire™ 6800/4810/4800/3800 System Controller Command Reference Manual

---

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054 U.S.A.  
650-960-1300

Part No. 816-4852-10  
October 2002, [Revision A](#)

Send comments about this document to: [docfeedback@sun.com](mailto:docfeedback@sun.com)

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, OpenBoot, Sun Fire, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the Sun Microsystems, Inc. license agreements and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct. 1998), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuelle relatant à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, OpenBoot, Sun Fire, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licences de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



# Contents

---

**Preface** xiii

- 1. Introduction to the System Controller** 1
  - Platform and Domain Administration 1
  - Connecting to the System Controller 2
  - System Controller Navigation 2
  - Terminating Sessions 2
  
- 2. System Controller Syntax, Arguments, and Device Names** 3
  - Command Syntax and Arguments 3
  - Command Names 4
  - Component Names 4
  - Board States for CPU/Memory Boards and I/O Assemblies 6
  - Board Test Status 6
  - Domain Status 7
  - Using FTP URLs 8
  - Interactive Commands 9
  - Context-Sensitive Commands 9
  - Command Line Editing 10
  - Command Line History 11

Command Completion 12

### **3. System Controller Alphabetical Command Reference 13**

Command Summary 13

Alphabetical Listing of System Controller Commands 16

addboard 16  
addcodlicense 19  
break 21  
connections 23  
console 25  
deleteboard 27  
deletecodlicense 29  
disablecomponent 31  
disconnect 35  
dumpconfig 37  
enablecomponent 39  
flashupdate 42  
help 47  
history 49  
password 50  
poweroff 52  
poweron 55  
reboot 58  
reset 60  
restoreconfig 62  
resume 65  
setdate 67  
setdefaults 72  
setfailover 75

setkeyswitch	77
setupdomain	82
setupplatform	88
showboards	96
showcodlicense	102
showcodusage	104
showcomponent	108
showdate	111
showdomain	113
showenvironment	117
showerrorbuffer	122
showfailover	125
showkeyswitch	128
showlogs	130
showplatform	132
showresetstate	138
showsc	140
testboard	142

**Glossary 145**

**Index 147**



# Tables

---

TABLE 2-1	Components That Are Accessible From the Platform	5
TABLE 2-2	Components That Are Accessible From a Domain	5
TABLE 2-3	Board States	6
TABLE 2-4	Status Field of the <code>showboards</code> Command	6
TABLE 2-5	Values for the Domain Status Column in <code>showplatform</code> and <code>showdomain</code> Command Output	7
TABLE 2-6	Command Line Editing Capabilities	10
TABLE 2-7	Keystrokes that Recall Previously Typed Command Lines	11
TABLE 2-8	Keystrokes Used to Complete System Controller Commands	12
TABLE 3-1	System Controller Command Summary	13
TABLE 3-2	Header Description for the <code>connections</code> Command	24
TABLE 3-3	<i>component_name</i> Descriptions for the CPU/Memory Board Used in the <code>disablecomponent</code> Command	32
TABLE 3-4	<i>component_name</i> Descriptions for the I/O Assembly Used in the <code>disablecomponent</code> Command	32
TABLE 3-5	<i>component_name</i> Descriptions for the CPU/Memory Board Used in the <code>enablecomponent</code> Command	39
TABLE 3-6	<i>component_name</i> Descriptions for the I/O Assembly Used in the <code>enablecomponent</code> Command	40
TABLE 3-7	Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time	68
TABLE 3-8	Description of the <code>setkeyswitch</code> Transition	78
TABLE 3-9	Results of Changing the Keyswitch From the Current Setting to a New Setting	78

TABLE 3-10	Parameter Values for the <code>setupdomain</code> Command	82
TABLE 3-11	<code>setupplatform</code> Parameter Values	89
TABLE 3-12	Output Header Definitions for the <code>showboards</code> and the <code>showboards -v</code> Command	98
TABLE 3-13	COD License Information	102
TABLE 3-14	<code>showcodusage</code> Resource Information	105
TABLE 3-15	<code>showcodusage</code> Domain Information	106
TABLE 3-16	Parts for <code>showenvironment -p part</code>	118
TABLE 3-17	<code>showenvironment</code> Output Header Description	118



# Code Samples

---

CODE EXAMPLE 3-1	<code>addboard</code> Example in the Platform Shell	18
CODE EXAMPLE 3-2	<code>addboard</code> Example in the Domain Shell	18
CODE EXAMPLE 3-3	<code>addcodlicense</code> Command Example	20
CODE EXAMPLE 3-4	<code>break</code> Command Example	22
CODE EXAMPLE 3-5	<code>connections</code> Command for the Platform Shell	24
CODE EXAMPLE 3-6	<code>console</code> Example and No Password Set Accessing Domain B	26
CODE EXAMPLE 3-7	<code>console</code> Example and A Password Set Accessing Domain A	26
CODE EXAMPLE 3-8	<code>deleteboard</code> Example Showing Deleting I/O Assembly 7	28
CODE EXAMPLE 3-9	<code>deleteboard</code> Example Showing Deleting CPU/Memory Board 3	28
CODE EXAMPLE 3-10	<code>deletecodlicense</code> Command Example	30
CODE EXAMPLE 3-11	<code>disablecomponent</code> Command Example Adding <code>sb4</code> to the Blacklists	34
CODE EXAMPLE 3-12	<code>disablecomponent</code> Command Example Adding <code>sb0</code> CPU Port 3 to the Blacklist	34
CODE EXAMPLE 3-13	<code>disablecomponent</code> Command Example Adding <code>sb2</code> , CPU Port 3, and Physical Bank 0 to the Blacklists	34
CODE EXAMPLE 3-14	<code>disablecomponent</code> Command Example Adding I/O Assembly 9, Port 0 and Bus 1 and Also I/O Assembly 8, I/O Card 2 to the Blacklists	34
CODE EXAMPLE 3-15	Disconnecting a Session From the System Controller Platform Shell	36
CODE EXAMPLE 3-16	Disconnecting a Session From the System Controller Platform Console	36
CODE EXAMPLE 3-17	<code>disconnect</code> Example Showing Disconnecting a Session From the Domain A Shell	36
CODE EXAMPLE 3-18	<code>dumpconfig</code> Example	38
CODE EXAMPLE 3-19	<code>enablecomponent</code> Command Example Enabling CPU/Memory Board 4	41

CODE EXAMPLE 3-20	<code>enablecomponent</code> Command Example Enabling I/O Assembly 6 and Port 1	41
CODE EXAMPLE 3-21	<code>flashupdate</code> Example Updating the Active System Controller Board and the System Controller Real Time Operating System	46
CODE EXAMPLE 3-22	<code>flashupdate</code> Example Updating Replacement CPU/Memory Board <code>sb4</code>	46
CODE EXAMPLE 3-23	<code>help</code> Command Example Displaying Information on the <code>addboard</code> Command	48
CODE EXAMPLE 3-24	<code>help</code> Command Example Displaying Commands Beginning With the Word <code>show</code>	48
CODE EXAMPLE 3-25	<code>help</code> Command Example Displaying Commands Beginning With the Letter A	48
CODE EXAMPLE 3-26	<code>history</code> Command	49
CODE EXAMPLE 3-27	<code>password</code> Command Example With a Password Already Set	51
CODE EXAMPLE 3-28	<code>password</code> Command Example With No Password Set	51
CODE EXAMPLE 3-29	<code>poweroff</code> Command Example Showing Powering Off <code>sb2</code>	54
CODE EXAMPLE 3-30	<code>poweroff</code> Command Example Showing Powering Off All Power Supplies, Fan Trays, and System Boards in the Platform Shell	54
CODE EXAMPLE 3-31	<code>poweron</code> Example Powering On <code>sb2</code>	56
CODE EXAMPLE 3-32	<code>poweron</code> Example Showing Powering On All CPU/Memory Boards and I/O Assemblies in Domain A	57
CODE EXAMPLE 3-33	<code>reboot</code> Command Example	59
CODE EXAMPLE 3-34	<code>reset</code> Command Example From Domain A	61
CODE EXAMPLE 3-35	<code>reset -a</code> Command Example From Domain A	61
CODE EXAMPLE 3-36	<code>restoreconfig</code> Example	63
CODE EXAMPLE 3-37	<code>resume</code> Command Example in Active Domain A	66
CODE EXAMPLE 3-38	<code>setdate</code> Command Example in the Platform Shell	70
CODE EXAMPLE 3-39	<code>setdate -r</code> Command Example Setting the Date From a Date Host	70
CODE EXAMPLE 3-40	<code>setdate -t</code> Command Example Setting the Time Zone to Pacific Standard Time Using the Offset From Greenwich Mean Time	70
CODE EXAMPLE 3-41	<code>setdate -t</code> Command Example Setting the Date and Time Zone to Eastern Standard Time Using Time Zone Abbreviations	70
CODE EXAMPLE 3-42	<code>setdate -t</code> Command Example Setting the Time Zone to Japan Standard Time Using the Offset From Greenwich Mean Time	71
CODE EXAMPLE 3-43	<code>setdate -t</code> Command Example Setting the Time Zone to Eastern Central Time Using Time Zone Abbreviations	71
CODE EXAMPLE 3-44	<code>setdefaults</code> Example	74
CODE EXAMPLE 3-45	<code>setfailover</code> Command Example (run on the Spare SC) of Manual Failover	76

CODE EXAMPLE 3-46 `setfailover` Command Example Disabling Failover 76

CODE EXAMPLE 3-47 `setfailover` Command Example Re-enabling Failover 76

CODE EXAMPLE 3-48 `setkeyswitch on` Example 80

CODE EXAMPLE 3-49 `setkeyswitch off` Example 81

CODE EXAMPLE 3-50 `setkeyswitch off` Example Shutting Down Domain B From the Platform Shell 81

CODE EXAMPLE 3-51 Variables for the `setupdomain` Command 86

CODE EXAMPLE 3-52 `setupplatform` Output 94

CODE EXAMPLE 3-53 `showboards` Command for the Platform Shell 97

CODE EXAMPLE 3-54 `showboards` Command with the `-p version` Option 98

CODE EXAMPLE 3-55 `showboards -a` Command for the Domain Shell 101

CODE EXAMPLE 3-56 `showcodlicense` Example Output – Formatted COD RTU License Data 103

CODE EXAMPLE 3-57 `showcodlicense -r` Example Output – COD RTU License Keys 103

CODE EXAMPLE 3-58 `showcodlicense -v` Example Output – Formatted and Raw COD RTU License Data 103

CODE EXAMPLE 3-59 `showcodusage` Example Output by Resource 105

CODE EXAMPLE 3-60 `showcodusage` Example Output by Domain 106

CODE EXAMPLE 3-61 `showcodusage` Example Output by Both Resource and Domain 107

CODE EXAMPLE 3-62 `showcomponent sb4` Sample Output 109

CODE EXAMPLE 3-63 `showcomponent ib6` Sample Output 110

CODE EXAMPLE 3-64 `showdate` Command for the Platform Shell 112

CODE EXAMPLE 3-65 `showdomain` Command Example 114

CODE EXAMPLE 3-66 `showdomain -v` Command Example 115

CODE EXAMPLE 3-67 `showdomain -p bootparams` Example Displaying Boot Parameter Information 116

CODE EXAMPLE 3-68 `showenvironment` Platform Shell and Domain Shell Example 120

CODE EXAMPLE 3-69 `showerrorbuffer` Example Output for an Interconnect Test 123

CODE EXAMPLE 3-70 `showerrorbuffer` Example Output for a Hardware Error 123

CODE EXAMPLE 3-71 `showfailover` Command Indicating Failover is Functioning Normally 126

CODE EXAMPLE 3-72 `showfailover` Command Indicating Failover Has Been Disabled 126

CODE EXAMPLE 3-73 `showfailover` Command in Verbose Mode 126

CODE EXAMPLE 3-74 `showfailover` Command Indicating SC Failover is Inactive 127

CODE EXAMPLE 3-75	<code>showkeyswitch</code> Example Showing the Keyswitch Set to On	129
CODE EXAMPLE 3-76	<code>showkeyswitch</code> Example Showing the Keyswitch Set to Off	129
CODE EXAMPLE 3-77	<code>showkeyswitch</code> Example Showing the Keyswitch Set to Standby	129
CODE EXAMPLE 3-78	Sample Output of the <code>showlogs</code> Command Run After Rebooting the System	131
CODE EXAMPLE 3-79	<code>showplatform</code> Output for a Sun Fire 6800 System	134
CODE EXAMPLE 3-80	<code>showplatform -v</code> Output for a Sun Fire 6800 System	135
CODE EXAMPLE 3-81	<code>showsc</code> Command	140
CODE EXAMPLE 3-82	<code>testboard</code> Example Testing CPU/Memory Board <code>sb0</code>	143

# Preface

---

This book describes the Sun Fire™ 6800/4810/4800/3800 system controller command line interface. The system controller is responsible for controlling system functions, environmental monitoring, domain control, and hardware control of the server. Using the command line interface, you can configure the platform and domains, power on and off domains, power supplies, fans, and other components. Access to the Solaris domain console is also possible using the command line interface.

---

## Before You Read This Book

This book is written for system administrators or people who have a working knowledge of the Solaris™ operating environment. If you do not have such knowledge, you should first read the Solaris User and System Administrator document collections and consider UNIX® system administration training. For the documentation URLs, see “Accessing Sun Documentation Online” on page xv.

---

## How This Book Is Organized

This book contains the following chapters:

Chapter 1 introduces you to the system controller and briefly describes platform administration and domain administration.

Chapter 2 explains system controller command syntax, command names, and command arguments.

Chapter 3 provides a summary in tabular form of all of the system controller commands and describes each command.

---

## Typographic Conventions

Typeface <sup>1</sup>	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
<b>AaBbCc123</b>	What you type, when contrasted with on-screen computer output	% <b>su</b> Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this. To delete a file, type <code>rm filename</code> .

<sup>1</sup> The settings on your browser might differ from these settings.

---

## Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

---

## Related Documentation

Application	Title	Part Number
Service	<i>Sun Fire 6800/4810/4800/3800 Systems Service Manual</i>	805-7363
Service	<i>Sun Fire 4810/4800/3800 System Cabinet Mounting Guide</i>	806-6781
System Administration	<i>Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual</i>	816-4851-10
System Administration	<i>Sun Fire 6800/4810/4800/3800 Systems Firmware 5.14.0 Release Notes</i>	816-4853-10

---

## Accessing Sun Documentation Online

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

<http://www.sun.com/documentation>

---

## Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

[docfeedback@sun.com](mailto:docfeedback@sun.com)

Please include the part number (816-4852-10) of your document in the subject line of your email.





# Introduction to the System Controller

---

The system controller consists of the System Controller board and the system controller software.

The system controller software does the following:

- Monitors and controls the system
- Manages hardware
- Configures domains
- Provides the date and time to the Solaris operating environment
- Provides the clock signal used on all system boards
- Provides a platform console and a domain console
- Provides system monitoring and control using SNMP for use with the Sun Management Center 3.0 software.

For more information on the system controller, refer to the “Overview” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* and also the *Sun Fire 6800/4810/4800/3810 Systems Overview Manual*.

---

## Platform and Domain Administration

The platform administration function manages resources and services that are shared among the domains. With this function, you can configure how resources and services are configured and shared.

The domain administration function manages resources and services for a specific domain. With this function you can configure the domain, control the keyswitch position, and access the domain console.

For more information on the platform administration and domain administration functions, refer to the “Overview” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

---

## Connecting to the System Controller

To connect to the system controller, refer to the “System Controller Navigation Procedures” chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

---

## System Controller Navigation

To navigate within the system controller, refer to the “System Controller Navigation Procedures” chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

---

## Terminating Sessions

To terminate system controller sessions, refer to the “System Controller Navigation Procedures” chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

# System Controller Syntax, Arguments, and Device Names

---

This chapter describes the following topics:

- “Command Syntax and Arguments” on page 3
- “Command Names” on page 4
- “Component Names” on page 4
- “Board States for CPU/Memory Boards and I/O Assemblies” on page 6
- “Board Test Status” on page 6
- “Domain Status” on page 7
- “Using FTP URLs” on page 8
- “Interactive Commands” on page 9
- “Context-Sensitive Commands” on page 9
- “Command Line Editing” on page 10
- “Command Line History” on page 11
- “Command Completion” on page 12

---

## Command Syntax and Arguments

The general syntax of system controller commands is:

*command\_name* [ *flags* ] [ *arguments* ]

Arguments are the words that follow the command name and are divided into two categories:

- Required arguments
- Optional arguments

Optional arguments always follow the option flags on the command line. In the following example, the optional *domainID* argument, *a*, follows the option flag, *-d*. The *boardname* argument, which is *sb2* in this example, is a required argument and is the board name for a CPU/Memory board.

```
schostname:SC> addboard -d a sb2
```

---

## Command Names

Most system controller commands are in the form of *verbnoun*. Command names are in the English language. System controller commands names are case insensitive, but options for commands are not. However, items such as board names can be specified in upper case characters.

---

**Note** – When typing system controller commands, you can type the complete command name or type just enough of the command name to uniquely identify it.

---

---

## Component Names

TABLE 2-1 shows the component names that are accessible from the platform. TABLE 2-2 shows the component names that are accessible from the domain. In both tables, you can type the component names in either upper case or lower case.

---

**Note** – The components in TABLE 2-1 and TABLE 2-2 depend on the system you have. For example, only the Sun Fire 6800 system can have six power supplies and six CPU/Memory boards.

---

**TABLE 2-1** Components That Are Accessible From the Platform

Device Description	Device Name
Power grids*	GRID0, GRID1
Power supplies	PS0, PS1, PS2, PS3, PS4, PS5
CPU/Memory boards	SB0, SB1, SB2, SB3, SB4, SB5
I/O assemblies	IB6, IB7, IB8, IB9
Repeater boards	RP0, RP1, RP2, RP3
ID/Source board	ID0
Fan trays	FT0, FT1, FT2, FT3
System controller	SSC0, SSC1

\* Power grids are not a component but a division of the Sun Fire 6800 system into two distinct halves of the system. Power supplies ps0, ps1, and ps2 comprise power grid 0. Power supplies ps3, ps4, and ps5 comprise power grid 1.

TABLE 2-2 shows the components that are accessible from a domain.

**TABLE 2-2** Components That Are Accessible From a Domain

Device Description	Device Name
CPU/Memory boards	SB0, SB1, SB2, SB3, SB4, SB5
I/O Assemblies	IB6, IB7, IB8, IB9

---

# Board States for CPU/Memory Boards and I/O Assemblies

TABLE 2-3 lists the board states for the CPU/Memory boards and I/O assemblies. To determine the board state, use the `showboards` command and look under the `State` header.

**TABLE 2-3** Board States

Board States	Description
Available	The board is not assigned to any domain.
Assigned	The board belongs to a domain, but the hardware has not been configured or it is not in use.
Active	The board is being actively used by the domain to which it has been assigned. You cannot reassign an active board.

If you have redundant system controllers, the SC state is identified as either the `Main` or `Spare`. The main SC provides all system resources, while the spare SC can provide system resources if the main SC fails.

The ID board, power supplies, and Repeater boards do not have a state. A dash is displayed in the Board Status field for these boards and components (TABLE 2-4). Fan trays have a state of off, low speed, and high speed.

---

# Board Test Status

TABLE 2-4 describes the Status field of the `showboards` command.

**TABLE 2-4** Status Field of the `showboards` Command

Test Status	Description
Passed	All board components passed testing.
Failed	The board failed testing and is not usable. This could also indicate corrupt or incompatible firmware.
OK	The component is functioning properly.

**TABLE 2-4** Status Field of the `showboards` Command (Continued)

Test Status	Description
Under Test	The domain is running POST (power-on self-test); testing has been initiated using the <code>setkeyswitch</code> command, the <code>testboard</code> command, or an SNMP operation.
Not Tested	The board has not been tested.
Degraded	Certain components on the board have failed or are disabled. A board is degraded when there are still usable parts on the board.
-	The slot is empty or not applicable for this device.

## Domain Status

In the `showplatform` and `showdomain` commands, one of the fields in the command output is domain status. The main values of domain status are (TABLE 2-5):

**TABLE 2-5** Values for the Domain Status Column in `showplatform` and `showdomain` Command Output

Domain State	Description
Powered Off	The domain is not active and the keyswitch is in the off position.
Standby	The domain is not active and the keyswitch is in the standby position.
Running POST	The domain is active and is running POST (power-on self-test).
Active	The domain is active.
Active - OpenBoot PROM	The domain is active and is running the OpenBoot™ PROM.
Active - Booting	The domain is active and is booting the Solaris operating environment.
Active - Solaris	The domain is active and is running the Solaris operating environment.
Active - Halted	The Solaris operating environment is halted in the domain.
Active - Reset	The domain has had an XIR reset and had not been rebooted.
Active - Panicking	The domain is active and is panicking. It is creating the core file.
Active - Debugger	The domain is active and the debugger is presently running.
Not Responding	The domain is not responding.
Paused due to an error	The domain is paused due to a hardware error.

# Special Characters

To execute multiple commands on the same line, separate them with a semicolon (;). The following example executes both the `addboard` and `deleteboard` commands.

```
schostname:A> addboard sb2;deleteboard sb3
```

A pound sign (#) signifies the start of a comment on the current line. The following example executes the `addboard` command. Everything you type after the # and before pressing the Return key is ignored.

```
schostname:A> addboard sb2 #this text is ignored
```

---

## Using FTP URLs

When you use certain system controller commands, such as `flashupdate`, `dumpconfig`, or `restoreconfig`, where the URL uses the FTP protocol, specify absolute paths by typing a double slash (//) after the hostname. Otherwise, the path is interpreted relative to the home directory of the specified user.

The following examples show the various ways to specify an FTP URL:

- URL that uses an absolute path name:

```
ftp://user:password@hostname//tmp/directory
```

The example above references the `/tmp/directory`.

- URL that uses a relative path name:

```
ftp://user:password@hostname/tmp/directory
```

In the example above, the path name references `/home/user/tmp/directory`.

- URL with anonymous FTP (no user name or password):

```
ftp://hostname/tmp/directory
```



The path name in this example references `/home/ftp/tmp/directory`.

---

## Interactive Commands

Some commands may prompt for confirmation before executing the command. You can disable prompting by specifying the `-y` or `-n` flags, which answer yes or no to any prompted question.

Some commands, such as `setupplatform` and `setupdomain`, are always interactive.

- **When an interactive command prompts for input, do the following:**
  - Press the Return key to keep the current setting, displayed in brackets [ ].
  - Type a dash ( - ) to change the current setting to an empty string (if the input type is used to define a string).

---

## Context-Sensitive Commands

Be aware that certain system controller commands can be run on the main SC but not on the spare SC. For a list of the commands applicable to your SC, run the `help` command on your SC in the platform shell. The help listing identifies the valid commands, based on whether the system controller is the main or the spare.

---

# Command Line Editing

The system controller has a basic command line editor that allows you to edit the command line. The keys you press are *not* echoed on the screen.

TABLE 2-6 lists the keystrokes used to perform basic command line editing.

- To use the Control key sequences, hold down the Control key while typing the character that follows(TABLE 2-6).
- To use Escape key sequences, press and release the Escape key. Then press and release the following character (TABLE 2-6).

**TABLE 2-6** Command Line Editing Capabilities

Keystroke	Description
Backspace (Control-h)	Erases the character before the cursor.
Delete	Erases the character before the cursor.
Control-c	Cancels editing the command line. The command is not executed.
Return	Finishes editing the line.
Control-b	Moves backward one character.
Esc-b	Moves backward one word.
Control-f	Moves forward one character.
Esc-f	Moves forward one word.
Control-a	Moves backward to the beginning of the line.
Control-e	Moves forward to the end of the line.
Esc-h	Erases from the beginning of the word to just before the cursor. Stores erased characters in a save buffer.
Control-w	Erases from the beginning of the line to just before the cursor. Stores erased characters in a save buffer.
Control-d	Erases next character.
Esc-d	Erases from the cursor to the end of the word. Stores erased characters in a save buffer.
Control-k	Erases from the cursor to the end of the line. Stores erased characters in a save buffer.
Control-u	Erases the entire line. Stores erased characters in a save buffer.

**TABLE 2-6** Command Line Editing Capabilities (Continued)

Keystroke	Description
Control-r	Retypes the line.
Control-q	Quotes the next character. Allows you to insert Control characters.
Control-y	Inserts the contents of the save buffer before the cursor.

---

## Command Line History

Each shell keeps a short history of all entered commands. To display the command line history, use the `history` command. After recalling the commands, you can either edit them or execute them by pressing the Return key. A history of at least eight previously typed command lines are saved.

TABLE 2-7 lists keystrokes you type to recall previously typed commands.

**TABLE 2-7** Keystrokes that Recall Previously Typed Command Lines

Keystroke	Description
Control-p	Selects and displays the previous line for subsequent editing.
Control-n	Selects and displays the next line for subsequent editing.
Control-l	Displays the entire command history list.

---

# Command Completion

Using the command completion function makes it easier for you to enter long commands. After typing part of the command, use the completion keystroke (TABLE 2-8). This action searches the list of available commands for commands beginning with the characters you typed thus far.

- If there is only one possible command, the remainder of the characters are entered for you automatically.
- If there are several possibilities, any characters that are common to all of the possible selections are filled in.
- If there are no defined commands starting with the given characters, the characters are erased until there is a minimum of one selection for the remaining characters.

TABLE 2-8 lists the keystrokes to complete a command.

**TABLE 2-8** Keystrokes Used to Complete System Controller Commands

Key Sequence	Description
Control-space	Complete this command.
Control-?	Shows all possible matches.
Control-/	Shows all possible matches.

# System Controller Alphabetical Command Reference

This chapter provides a summary of the system controller commands designed for an end-user on the Sun Fire 6800/4810/4800/3800 systems, describes the system controller command line interface, and provides descriptions and examples for each command.

---

## Command Summary

TABLE 3-1 lists and describes the system controller commands and how you can access them. Many commands are accessible from both the platform shell and the domain shell. The system controller commands may differ in how they are used, the effect of the command, and the scope of the command between the platform and domain shells.

**TABLE 3-1** System Controller Command Summary

Command	Description	Platform Shell	Domain Shell
addboard	Assigns a board to a domain.	x	x
addcodlicense	Adds a Capacity on Demand (COD) right-to-use (RTU) license key to the COD license database.	x	
break	Sends a Break signal to the domain console.		x
connections	Displays connections to the system controller to a domain.	x	x
console	Connects to a domain from the platform.	x	
deleteboard	Unassigns a board from a domain.	x	x

**TABLE 3-1** System Controller Command Summary *(Continued)*

<b>Command</b>	<b>Description</b>	<b>Platform Shell</b>	<b>Domain Shell</b>
deletecodlicense	Removes a Capacity on Demand (COD) right-to-use (RTU) license key from the COD license database.	x	
disablecomponent	Adds a component to the blacklist.	x	x
disconnect	Disconnects the current or specified connection.	x	x
dumpconfig	Saves the system controller configuration to a server.	x	
enablecomponent	Deletes a component from the blacklist.	x	x
flashupdate	Updates the firmware.	x	
help	Provides basic help information for commands.	x	x
history	Shows the command history with date and time stamps.	x	x
password	Sets the shell password.	x	x
poweroff	Powers off components.	x	x
poweron	Powers on components.	x	x
reboot	Reboots the system controller.	x	
reset	Resets the domain in the domain shell.		x
restoreconfig	Restores the system controller configuration from a server.	x	
resume	Exits the domain shell and resumes access to the domain console.		x
setdate	Sets the date and time.	x	x
setdefaults	Sets the configuration to default values.	x	x
setfailover	Changes the state of system controller failover.	x	
setkeyswitch	Sets the keyswitch position.	x	x
setupdomain	Configures the domain.		x
setupplatform	Configures the platform.	x	
showboards	Shows board information.	x	x
showcodlicense	Displays the current Capacity on Demand (COD) right-to-use (RTU) licenses stored in the COD license database.	x	
showcodusage	Displays the current usage statistics for Capacity on Demand (COD) resources.	x	x
showcomponent	Shows state of a component.	x	x

**TABLE 3-1** System Controller Command Summary *(Continued)*

<b>Command</b>	<b>Description</b>	<b>Platform Shell</b>	<b>Domain Shell</b>
showdate	Shows the date and time.	x	x
showdomain	Displays the configuration and status of the domain.		x
showenvironment	Shows environmental information.	x	x
showerrorbuffer	Shows the contents of the error buffer.	x	
showfailover	Displays system controller and clock failover status.	x	
showkeyswitch	Displays the virtual keyswitch setting.	x	x
showlogs	Displays the logs.	x	x
showplatform	Displays the configuration and information for the platform and status of the domains.	x	
showresetstate	Shows CPUs after a reset or a RED MODE trap.		x
showsc	Shows the system controller version and uptime.	x	
testboard	Tests the CPU/Memory board.	x	x

---

# Alphabetical Listing of System Controller Commands

The following sections describe the system controller commands.

## addboard

Assigns a board to a domain.

### Scope

platform shell, domain shell

### Syntax for the Platform Shell

```
addboard -d domainID system_board_name [system_board_name . . .]
```

```
addboard -h
```

### Syntax for the Domain Shell

```
addboard system_board_name [system_board_name . . .]
```

```
addboard -h
```

### Options/Parameters

`-d` specifies a *domainID* as an option (platform shell only).

*domainID* is a, b, c, or d.

`-h` displays help for this command.

*system\_board\_name* is the board to be added. Possible values for *system\_board\_name* are sb0-sb5 (CPU/Memory board) or ib6 - ib9 (I/O assembly).



## Description

Assigns *system\_board\_name* to the specified domain within the platform shell or to the current domain. The board state must be *Available*. To display the board state, use the `showboards` command. For more information on board states, see “Board States for CPU/Memory Boards and I/O Assemblies” on page 6.

To assign a board to a domain using a domain shell, the board name must be listed in the access control list (ACL) for the current domain. When a board is listed in the ACL, the system controller software is allowed to process `addboard` requests on that board. The platform shell does not use the ACL and can always add boards to a domain. The platform shell can also delete boards from a domain and either ignores the ACL or overrides the ACL. You configure the ACLs and also restrict the domains a board has access to using the `setupplatform` command.

If a board is assigned to an active domain, the board will not be used. To unassign a board from a domain, you must halt the Solaris operating environment in the domain. For an overview of steps to perform, see the section “Assigning and Unassigning Boards” in the “Maintenance” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

If a board is not present, the command assigns ownership of the slot to the specified domain.

## See Also

`deleteboard`, `setkeyswitch`, `showboards`, `showdomain`, `showplatform`, “Assigning and Unassigning Boards” in the “Maintenance” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for a step-by-step procedure on assigning boards to a domain)

## Example—Platform Shell

From the platform shell to add board name `sb2` to domain A (CODE EXAMPLE 3-1), type:

**CODE EXAMPLE 3-1** `addboard` Example in the Platform Shell

```
schostname:SC> addboard -d a sb2
```

## Example—Domain Shell

To assign board name `sb2` to the current domain (CODE EXAMPLE 3-2), type:

**CODE EXAMPLE 3-2** `addboard` Example in the Domain Shell

```
schostname:A> addboard sb2
```

# addcodlicense

Adds a Capacity on Demand (COD) right-to-use (RTU) license key to the COD license database.

## Scope

platform shell

## Syntax

```
addcodlicense license-signature
```

```
addcodlicense -h
```

## Options/Parameters

-h displays help for this command.

*license-signature* is the COD RTU license key to be added to the COD license database.

## Description

Adds the specified COD RTU license key to the COD license database on the system controller.

---

**Note** – Before you run this command, you must obtain a COD RTU license key from the Sun License Center. For details on COD RTU license keys, refer to the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

---

## See Also

deletecodlicense, showcodlicense, showcodusage

## Example

### CODE EXAMPLE 3-3 addcodlicense Command Example

```
schostname:SC> addcodlicense 01:80d8a9ed:45135285:0201000000:8:00000000:0000000000000000000000
```

---

**Note** – The COD RTU license key listed above is provided as an example and is not a valid license key.

---

# break

Sends a Break signal to the domain console.

## Scope

domain shell

## Syntax

```
break [-y|-n]
```

```
break -h
```

## Options/Parameters

-h displays help for this command.

-y answers yes to the warning message. Does not prompt for confirmation.

-n answers no to the warning message. Does not execute this command if confirmation is requested.

## Description

Sends a Break signal to the domain console. Resumes the domain console.

---

**Note** – This command pauses the Solaris operating environment.

---

When the Solaris operating environment is running in the domain, the usual effect of the `break` signal is to force entry into OpenBoot PROM or the debugger. The Solaris operating environment will ignore the break signal if the keyswitch is set to secure.

## See Also

`resume`, `setkeyswitch`, `showkeyswitch`

## Example

CODE EXAMPLE 3-4 shows using the `break` command to pause the Solaris operating environment and enter the OpenBoot PROM.

### CODE EXAMPLE 3-4 `break` Command Example

```
schostname:A> break
```

```
This will suspend Solaris in domain A.
```

```
Do you want to continue? [no] yes
```

```
Type 'go' to resume.
```

```
debugger entered.
```

```
{1} ok
```

# connections

Displays connections to the system controller or a domain.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
connections [-d domainID]
```

```
connections -h
```

## Syntax for the Domain Shell

```
connections [-h]
```

## Options/Parameters

-h displays help for this command.

-d *domainID* shows connections to the specified domain (a, b, c, or d).

## Description

Displays hosts currently connected to the system controller. When run from the platform shell, this command shows the connections to the platform and to each domain.

When run from the domain shell, this command will only show the connection to the domain. There can only be one connection to each domain.

## See Also

disconnect

## Example—Platform Shell

**CODE EXAMPLE 3-5** connections Command for the Platform Shell

```
schostname:SC> connections
ID      Hostname      Idle Time      Connected On      Connected To
--      -
1       XXXXXXXX     -              May 02 00:00     Platform
4       XXX          -              May 02 00:21     Platform

schostname:SC
```

TABLE 3-2 describes the headers in the output for CODE EXAMPLE 3-5.

**TABLE 3-2** Header Description for the connections Command

Header Column in the connections Command	Description
ID	Connection ID. This is a unique number.
Hostname	Source of the connection for this session. If the value is localhost, the connection was established through the platform or through the serial port.
Idle Time	Amount of time that the telnet or serial port connection has been idle, without any activity. Displays a dash ( - ) if there has been any activity during the last 30 seconds.
Connected On	Date and time the connection was initiated. The format is Mon dd hh:mm.
Connected To	Shell or console this session is connected to. Possible values are Platform or domainIDs A, B, C, or D.



# console

Connects to a domain from the platform.

## Scope

platform shell

## Syntax

```
console [-d] domainID
```

```
console -h
```

## Options/Parameters

`-d domainID` is the domain to connect to and is a, b, c, or d. The `-d` parameter is optional and does not need to precede the *domainID*.

`-h` displays help for this command.

## Description

Connect to a domain. You use the `console` command to navigate from the platform to a domain. If a password was created for connecting to this domain, you must enter the password.

If the domain is active (the domain is running the Solaris operating environment, OpenBoot PROM, or POST), you are connected to the domain console. Otherwise, you are connected to the domain shell. To obtain the domain shell from the domain console, see the chapter “System Controller Navigation Procedures” in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

## See Also

`disconnect`

## Examples—Platform Shell

When the domain is not active and there is no password initialized for this domain:

**CODE EXAMPLE 3-6** `console` Example and No Password Set Accessing Domain B

```
schostname:SC> console b

Connected to Domain B

Domain Shell for Domain B

schostname:B>
```

When a domain is not active and a password is initialized for this domain:

**CODE EXAMPLE 3-7** `console` Example and A Password Set Accessing Domain A

```
schostname:SC> console a

Enter Password:

Connected to Domain A

Domain Shell for Domain A

schostname:A>
```

When a domain is active and a password is not set for this domain:

```
schostname:SC> console a

Connected to Domain A
```

Note that no other output is seen. If the OpenBoot PROM is at the `ok` prompt or the Solaris operating environment is at the `login:` prompt, you must press the Enter key in order to see the prompt. Otherwise, you will need to wait for output to be generated by POST, the OpenBoot PROM, or the Solaris operating environment.

If the domain is hung, there will be no output. However, when a domain displays no output, this does not necessarily mean that the domain is hung.

# deleteboard

Unassigns a board from the domain where it is currently assigned.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell and the Domain Shell

```
deleteboard system_board_name [system_board_name . . .]
```

```
deleteboard [-h]
```

## Options/Parameters

-h displays help for this command.

*system\_board\_name* is the board to be deleted. Values are sb0 to sb5 (CPU/Memory board) and ib6 to ib9 (I/O assembly).

## Description

Unassigns a board from the domain where it is currently assigned. When you use this command from your current domain, you can only unassign boards that are assigned to the current domain. The CPU/Memory board or the I/O assembly board status must be in the *Assigned* state and the board must *not* be part of an active domain. To display the board states, use the `showboards` command. For more information on board states, see “Board States for CPU/Memory Boards and I/O Assemblies” on page 6. The board does *not* have to be in the access control list (ACL) to be unassigned from the domain.

To unassign an active board from a domain, you must halt the Solaris operating environment in the domain. Or, you can use DR to unassign an active board. For an overview of steps to perform, see the section “Assigning and Unassigning Boards” in the “Maintenance” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

If a board is not present, the command unassigns ownership of the slot from the specified domain.

## See Also

`addboard`, `showboards`, “Assigning and Unassigning Boards” in the “Maintenance” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* (for a step-by-step procedure on unassigning boards from a domain)

## Example—Platform Shell

To delete I/O assembly `ib7`, type:

**CODE EXAMPLE 3-8** `deleteboard` Example Showing Deleting I/O Assembly 7

```
schostname:SC> deleteboard ib7
```

## Example—Domain Shell

To delete CPU/Memory board `sb3` from the current domain, domain A, type:

**CODE EXAMPLE 3-9** `deleteboard` Example Showing Deleting CPU/Memory Board 3

```
schostname:A> deleteboard sb3
```

# deletecodlicense

Removes a Capacity on Demand (COD) right-to-use (RTU) license key from the COD license database.

## Scope

platform shell

## Syntax

```
deletecodlicense [-f] license-signature
```

```
deletecodlicense -h
```

## Options/Parameters

**-f** forces the specified COD RTU license key to be deleted from the COD license database, even if the license removal will result in a license violation.

**-h** displays help for this command.

*license-signature* is the COD RTU license key to be removed from the COD license database.

## Description

Removes a COD RTU license key from the COD license database on the system controller. For further information on COD RTU license keys, refer to the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

The system checks the number of licenses against the number of COD CPUs in use. If the license removal will result in an insufficient number of COD RTU licenses for the COD CPUs in use, the system will not delete the license key from the COD license database. If you want to delete the COD RTU license key, you must reduce the number of COD CPUs in use. You can either power off the appropriate number of domains or use dynamic reconfiguration (DR) to disconnect the appropriate number of boards.

## See Also

addcodlicense, showcodlicense, showcodusage

## Example

### CODE EXAMPLE 3-10 deletecodlicense Command Example

```
schostrname:SC> deletecodlicense 01:80d8a9ed:45135285:0201000000:8:00000000:0000000000000000000000
```

---

**Note** – The COD RTU license key listed above is provided as an example and is not a valid license key.

---

# disablecomponent

Adds a component to the blacklist.

## Scope

platform shell, domain shell

## Syntax

```
disablecomponent component_name [component_name ...]
```

```
disablecomponent -h
```

## Options/Parameters

-h displays help for this command.

*component\_name* is (see TABLE 3-3 and TABLE 3-4):

- *system\_board\_name/port/physical\_bank/logical\_bank* for the CPU/Memory board
- *system\_board\_name/port/bus* for an I/O assembly
- *system\_board\_name/card* for an I/O assembly

---

**Note** – You can use partial component names.

---

**TABLE 3-3** *component\_name* Descriptions for the CPU/Memory Board Used in the `disablecomponent` Command

Board or Device	Component Name
CPU system	<i>board_name/port/physical_bank/logical_bank</i>
CPU/Memory boards ( <i>system_board_name</i> )	SB0, SB1, SB2, SB3, SB4, SB5
Ports on the CPU/Memory board	P0, P1, P2, P3
Physical memory banks on CPU/Memory boards	B0, B1
Logical banks on CPU/Memory boards	L0, L1, L2, L3

**TABLE 3-4** *component\_name* Descriptions for the I/O Assembly Used in the `disablecomponent` Command

Board or Device	Component Name
I/O assembly system	<i>board_name/port/bus</i> or <i>board_name/card</i>
I/O assemblies ( <i>system_board_name</i> )	IB6, IB7, IB8, IB9
Ports on the I/O assembly	P0, P1
Buses on the I/O assembly	B0, B1
I/O cards in the I/O assembly	C0, C1, C2, C3, C4, C5, C6, C7—the number of cards varies with the I/O assembly



## Description

This command adds a component to the blacklist. Blacklisting provides a list of components that will not be tested and will not be configured into the domain when you change the domain's keyswitch from an inactive state to an active state or reboot the domain.

Blacklist a component or device if you believe it may be failing intermittently.

When you disable any component, the subcomponents are also disabled. For example, if you disable a CPU, the Ecache and memory that are also controlled by the CPU are automatically disabled.

Devices can be the following (see TABLE 3-3 and TABLE 3-4):

- Ports (CPU on a CPU/Memory board and I/O controller on an I/O assembly)
- Physical and logical memory banks
- I/O buses
- I/O cards

Blacklisting is done on a per domain basis. When you run the `disablecomponent` command in a domain shell, the component is blacklisted for the current domain. When you run the `disablecomponent` command from the platform shell, the component is disabled in all of the domains.

---

**Note** – If you are disabling ports on an I/O assembly, leave at least one I/O controller 0 enabled in a domain, so that the domain can communicate with the system controller.

---

Using this command, the platform shell blacklists supersede the domain shell blacklists. For example, if a component is disabled in the platform shell, it will be disabled in all domains.

## See Also

`enablecomponent`, `showcomponent`, and the “Disabling Components” section in the “Troubleshooting” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* for step-by-step procedure on adding a component to the blacklist.

## Examples

CODE EXAMPLE 3-11 adds *system\_board\_name* sb4 to the blacklist.

**CODE EXAMPLE 3-11** disablecomponent Command Example Adding sb4 to the Blacklists

```
schostname:A> disablecomponent sb4
```

CODE EXAMPLE 3-12 adds *system\_board\_name* sb0 and CPU port 3 to the blacklists. Note that any memory banks on this CPU port are unreachable and are implicitly disabled.

**CODE EXAMPLE 3-12** disablecomponent Command Example Adding sb0 CPU Port 3 to the Blacklist

```
schostname:A> disablecomponent sb0/p3
```

CODE EXAMPLE 3-13 adds *system\_board\_name* sb2, CPU port 3, physical bank 0 to the blacklists. Note that any logical banks belonging to this physical bank are disabled.

**CODE EXAMPLE 3-13** disablecomponent Command Example Adding sb2, CPU Port 3, and Physical Bank 0 to the Blacklists

```
schostname:A> disablecomponent sb2/p3/b0
```

CODE EXAMPLE 3-14 adds *system\_board\_name* ib9, port 0, bus 1 and *board\_name* ib8, I/O card 2 to the blacklists. Disabling I/O card 2 shuts down the power to that I/O card.

**CODE EXAMPLE 3-14** disablecomponent Command Example Adding I/O Assembly 9, Port 0 and Bus 1 and Also I/O Assembly 8, I/O Card 2 to the Blacklists

```
schostname:A> disablecomponent ib9/p0/b1 ib8/c2
```

# disconnect

Disconnects the current or specified connection.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
disconnect [ID]
```

```
disconnect -h
```

## Syntax for the Domain Shell

```
disconnect [-h]
```

## Options/Parameters

-h displays help for this command.

*ID* (platform shell only) is the connection to be terminated. *ID* is the number displayed under the heading ID for the `connections` command.

## Description

Terminates a connection to the system controller. If this command is used with no arguments, it disconnects the current session. For illustrations of the disconnect command, see the chapter “System Controller Navigation Procedures” in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

If the connection was initiated from another system, you will be returned to the remote host. If the session was initiated from the platform shell, you will see nothing. To obtain the prompt, press the Return key. If you are connected from the serial port, then you will return to the main menu.

## See Also

`connections`

## Examples

To disconnect a session from the system controller platform shell with a telnet connection (CODE EXAMPLE 3-15), type:

### CODE EXAMPLE 3-15 Disconnecting a Session From the System Controller Platform Shell

```
schostname:SC> disconnect  
Connection closed by foreign host.
```

If you are connected to the system controller platform console with the serial connection, you will see the following after typing `disconnect` (CODE EXAMPLE 3-16):

### CODE EXAMPLE 3-16 Disconnecting a Session From the System Controller Platform Console

```
schostname:SC> disconnect  
  
Type 0 for Platform Shell  
  
Type 1 for domain A  
Type 2 for domain B  
Type 3 for domain C  
Type 4 for domain D  
  
Input:
```

To disconnect a session from a system controller domain shell (CODE EXAMPLE 3-17), type:

### CODE EXAMPLE 3-17 `disconnect` Example Showing Disconnecting a Session From the Domain A Shell

```
schostname:A> disconnect  
Connection closed by foreign host.
```

CODE EXAMPLE 3-17 displays a direct connection to the domain and *is not* a connection to the domain made from the platform.

---

**Note** – When you disconnect from a domain shell, you will see nothing. Press the Return key to obtain the prompt.

---

# dumpconfig

Saves the platform and domain configurations to a server.

## Scope

platform shell

## Syntax

```
dumpconfig -f url
```

```
dumpconfig -h
```

## Options/Parameters

-h displays help for this command.

-f specifies the URL, which must use the ftp protocol. The URL must point to a directory and not a file. Examples are:

```
ftp://userid:password@host/path
```

```
ftp://host/path
```

For details on ftp URLs, see “Using FTP URLs” on page 8.

---

**Note** – The hostname you enter can be the hostname or a hostname with a fully qualified domain name. You cannot use a partial domain name. For example: *schostname*, *schostname.eng.sun.com* can be a valid hostname.

---

---

**Note** – The directory specified in the URL path must have write permission for the specified user. If a user was not specified, then anonymous ftp is used.

---

## Description

Saves the platform and domain configurations to a server for recovery. This command creates two data files at the specified URL: *schostname.nvci* and *schostname.tod*.

Use this command when you complete setting up the initial configuration of the platform and the domains, each time you modify the configuration or when you change the hardware configuration. For example, a configuration change occurs when you use any of the following commands: `setupplatform`, `setupdomain`, `setdate`, `addboard`, `deleteboard`, `enablecomponent`, `disablecomponent`, and `password`. By running the `dumpconfig` command again, the new platform and domain configurations are saved to a server.



---

**Caution** – This command should be run any time you change the platform or domain configuration (see the list of commands in the previous paragraph). Invoking this command is *very important* because if the System Controller board fails and you did not use this command to save the platform and domain configurations, you will have to manually reconfigure the platform and the domains.

---

If you need to replace the System Controller board in a single SC configuration, the configuration information saved through the `dumpconfig` command is used to restore the platform and domain configurations to the replacement System Controller board.

You *should not* use this command to revert to an old configuration. The `showplatform`, `showdomain`, `showcomponent`, and `showboards` system controller commands display all of the configuration and can be used as a reference.

## See Also

`restoreconfig`

## Example

### CODE EXAMPLE 3-18 `dumpconfig` Example

```
shostname:SC> dumpconfig -f ftp://hostname/path
Created: ftp://hostname/path/shostname.nvci
Created: ftp://hostname/path/shostname.tod
```

# enablecomponent

Deletes a component from the blacklist.

## Scope

platform shell, domain shell

## Syntax

```
enablecomponent component_name [component_name . . .]
```

```
enablecomponent -h
```

## Options/Parameters

-h displays help for this command.

*component\_name* is (TABLE 3-5 and TABLE 3-6)

- *system\_board\_name/port/physical\_bank/logical\_bank* for CPU/Memory boards
- *system\_board\_name/port/bus* for an I/O assembly
- *system\_board\_name/card* for an I/O assembly

**TABLE 3-5** *component\_name* Descriptions for the CPU/Memory Board Used in the enablecomponent Command

Board or Device	Component Name
CPU system	<i>board_name/port/physical_bank/logical_bank</i>
CPU/Memory boards ( <i>system_board_name</i> )	SB0, SB1, SB2, SB3, SB4, SB5
Ports on the CPU/Memory board	P0, P1 ,P2, P3
Physical memory banks on CPU/Memory boards	B0, B1
Logical banks on CPU/Memory boards	L0, L1, L2, L3

**TABLE 3-6** *component\_name* Descriptions for the I/O Assembly Used in the `enablecomponent` Command

Board or Device	Component Name
I/O assembly system	<i>board_name/port/bus</i> or <i>board_name/card</i>
I/O assemblies ( <i>system_board_name</i> )	IB6, IB7, IB8, IB9
Ports on the I/O assembly	P0, P1
Buses on the I/O assembly	B0, B1
I/O cards in the I/O assembly	C0, C1, C2, C3, C4, C5, C6, C7—the number of cards varies with the I/O assembly

## Description

Removes a component from the blacklist. Blacklisting provides a list of components that will not be tested and will not be configured into the domain when you change the keyswitch setting from an inactive state to an active state or when the domain is rebooted.

Components can be the following:

- Ports (CPU on the CPU/Memory board or I/O controller on the I/O assembly)
- Physical and logical memory banks
- I/O buses
- I/O cards

Blacklisting is done on a per domain basis. When you run the `enablecomponent` command from the domain, the component is removed from the blacklist for the current domain. When the `enablecomponent` command is run from the platform shell, the component is removed from the blacklist for all domains.

Using this command, the platform shell blacklists supersede the domain shell blacklists. For example, if a component is enabled in the platform shell, it will be enabled in all domains.



## See Also

`disablecomponent`, `showcomponent`, and the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* for step-by-step procedure on removing a component from the blacklist. This information is in the “Troubleshooting” chapter in the section “Disabling Components.”

## Examples

CODE EXAMPLE 3-19 enables CPU/Memory board `sb4`. This removes CPU/Memory board `sb4` from the blacklist.

**CODE EXAMPLE 3-19** `enablecomponent` Command Example Enabling CPU/Memory Board 4

```
schostname:A> enablecomponent sb4
```

CODE EXAMPLE 3-20 enables I/O assembly 6, port 1. This removes port 1 of I/O assembly 6 from the blacklist.

**CODE EXAMPLE 3-20** `enablecomponent` Command Example Enabling I/O Assembly 6 and Port 1

```
schostname:A> enablecomponent ib6/p1
```

# flashupdate

Updates the firmware on the system controller and the system boards (CPU/Memory boards and I/O assemblies). The source flash image can be on a server or another board of the same type.

## Scope

platform shell

## Syntax

```
flashupdate [-y|-n] -f url all|systemboards|scapp|rtos
flashupdate [-y|-n] -f url board [board . . . ]
flashupdate [-y|-n] -u
flashupdate [-y|-n] -c source_board destination_board [destination_board . . . ]
flashupdate -h
```

## Options/Parameters

-h displays help for this command.

-y does not prompt for confirmation.

-n does not execute this command if confirmation is required.

-f specifies a URL as the source of the flash images

*url* is the URL to the directory containing the flash images.

where:

`ftp://hostname/path`

`ftp://userid:password@hostname/path`

`http://hostname/path`

For details on ftp URLs, see “Using FTP URLs” on page 8.

---

**Note** – The hostname you enter must be a host name or a host name with a fully qualified domain name. You cannot use a partial domain name. Some examples include: *schostrname*, *schostrname.eng.sun.com*.

---

*board* is the board name.

-c specifies that the image should be copied from another board.

- *source\_board* is the source board for the flash images.
- *destination\_board* is the destination board for the flash images.

-u automatically updates all of the boards from the board with the highest revision.

all updates the system controller, all system boards (CPU/Memory boards and I/O assemblies), and the system controller real time operating system (*rtos*).

*system\_boards* are all CPU/Memory boards and I/O assemblies.

*scapp* updates the current system controller. Updating the system controller reboots the system controller and disconnects all the current connections. If you have a second System Controller board installed, when you run *scapp*, also update *scapp* on the second System controller board.

*rtos* updates the real time operating system for the system controller. If you have a second System Controller board installed, when you update the real time operating system also update the *rtos* on the second System Controller board.

## Description



---

**Caution** – Any time the firmware is upgraded, check the *Install.info* file for the firmware upgrade procedure. This file is provided with your latest software release. Also refer to the *Release Notes* for any notes or special procedures. When you update the firmware on the system controller, update only one system controller at a time. DO NOT update both system controllers at the same time.

---

Updates the firmware on the system controller, CPU/Memory boards, and I/O assemblies. There is no firmware on Repeater boards. Boards must be powered on in order to be updated with *flashupdate*. If a board does not have power, the *flashupdate* command will do nothing and stop with an error message. Boards that are in a secure domain *cannot* be updated.

If you install a replacement board into the system:

1. Check that the firmware level of the replacement board is the same as the board you replaced. Incompatible boards are noted by the *Failed* board status in the output of the *showboards* command.

2. If the board level is not the same as the board you replaced, perform the `flashupdate` procedure on the board.
3. If a board was noted by the `Failed` state in `showboards`, after you `flashupdate` a compatible version, power off the board to clear the `Failed` state.

Before you update the flash images, make sure the firmware on the replacement board is compatible with the firmware of the board you are replacing for `scapp`, CPU/Memory board, and I/O assembly PROMs. New firmware is loaded if you reboot the domain or turn the domain keyswitch off with `setkeyswitch off` and then turn it on with `setkeyswitch on`.

- To determine the version number of the current firmware, use the `showboards -p version` command.

The new firmware may also contain a new `rtos` image. If the `rtos` image that is currently installed is different than the `rtos` supplied with the software, you must upgrade the `rtos` image.

- To determine the version number of the current `rtos`, use the `showsc` or the `showboards -v -p version` command. The `README` file also contains the version number of the new `rtos` image.

---

**Note** – If you fail to check all versions for compatibility, the domains can crash or the system controller can fail to reboot normally.

---



---

**Caution** – While upgrading the firmware, heed the following precautions:

- Read the `README` and `Install.info` files before you upgrade the firmware.
  - *Do not* change the keyswitch position of any domain.
  - *Do not* use DR.
  - *Do not* power off any boards.
  - *Do not* reboot the system controller.
- 

If you upgrade the `scapp` or `rtos` images for status:

1. It is important to watch the console during the `flashupdate` procedure.
2. Monitor the console if errors are reported.

If the images that are already installed are *incompatible* with the new images:

1. Shut down the domains before performing the firmware upgrade.
2. After you perform the firmware upgrade, reboot the domains.

If the images that are already installed are *compatible* with the new images:

1. Reboot the domains after performing the `flashupdate` procedure. Even though the firmware is compatible with the older firmware version, you must upgrade the firmware to obtain new features and bug fixes.

If you install a replacement board into the system:

1. Check if the firmware level of the replacement board is the same as the board you replaced.
2. If it is not, perform the `flashupdate` procedure on the board.



---

**Caution** – Under normal circumstances, the `flashupdate` command will complete successfully. However, if the `flashupdate` command is terminated abnormally (such as a power failure, a failed network connection, and so on), the system controller will prompt you for the URL of the images to be installed.

---

## See Also

*Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*

## flashupdate Command Platform Shell—Examples

---

**Note** – In the following examples, since the output is very long, only the command you type is listed in the code box.

---

---

**Note** – The `flashupdate all`, `flashupdate rtos`, and `flashupdate scapp` commands cause the system controller to reboot once.

---

- To update the active System Controller board, all the system boards, and the system controller real time operating system (`rtos`), perform this procedure from the platform console and watch the console output. This command reboots the system controller. Type:

**CODE EXAMPLE 3-21** `flashupdate` Example Updating the Active System Controller Board and the System Controller Real Time Operating System

```
schostname:SC> flashupdate -f ftp://host/path all
```

- To upgrade the firmware on a replacement CPU/Memory board, `sb4` (`destination_board`), which is a replacement CPU/Memory board for CPU/Memory board, `sb0` (`source_board`). Type:

**CODE EXAMPLE 3-22** `flashupdate` Example Updating Replacement CPU/Memory Board `sb4`

```
schostname:SC> flashupdate -c sb0 sb4
```

# help

Provides basic help information for commands.

## Scope

platform shell, domain shell

## Syntax

```
help [command_name] | [partial_command_name]
```

```
help -h
```

## Options/Parameters

`-h` displays help for this command.

*command\_name* is the name of the command.

*partial\_command\_name* can be one letter of the command or a portion of the command name, such as `show`.

## Description

The `help` command, without arguments, lists currently available commands. When an argument is supplied, the `help` command displays a list of commands that begin with the specified argument. If only one command is found, full help is displayed. Otherwise, a short description is displayed for each command, beginning with the specified argument.

You can also type a partial command name and basic help information will be returned for all commands matching the partial name. The partial command name must contain enough of the command name in order for the command to be recognized by `help`. The `help` command returns help information for all commands beginning with the characters you typed.

The `help` command is shell sensitive. It displays help information for the current shell only.

## Examples

CODE EXAMPLE 3-23 displays help information on the `addboard` command.

**CODE EXAMPLE 3-23** `help` Command Example Displaying Information on the `addboard` Command

```
shostname:SC> help addboard

addboard -- assign a board to a domain

Usage: addboard -d <domain> <board> ...
       addboard -h

       -d -- the domain to assign the board to
       -h -- display this help message
```

CODE EXAMPLE 3-24 displays all commands in the platform shell beginning with `show`.

**CODE EXAMPLE 3-24** `help` Command Example Displaying Commands Beginning With the Word `show`

```
shostname:SC> help show

showboards      -- show board information
showcodlicense  -- show COD licenses
showcodusage    -- show COD resource usage
showcomponent   -- show state of a component
showdate        -- show the current date and time for the platform
showenvironment -- show environmental information
showerrorbuffer -- show the contents of the error buffer
showfailover    -- show SC Failover information
showkeyswitch   -- show the keyswitch positions
showlogs        -- show the logs
showplatform    -- show platform configuration, information and domain status
showsc          -- show system controller version and uptime
```

CODE EXAMPLE 3-25 displays all commands in the platform shell beginning with the letter `a`.

**CODE EXAMPLE 3-25** `help` Command Example Displaying Commands Beginning With the Letter `A`

```
shostname:SC> help a

addboard      -- assign a board to a domain
addcodlicense -- add a cod license
```



# history

Shows the command history with date and time stamps.

## Scope

platform shell, domain shell

## Syntax

```
history [-h]
```

## Options/Parameters

-h displays help for this command.

## Description

Shows the command history with date and time stamps for when the commands were executed. This command is shown for your shell and displays the last twenty commands only.

## See Also

Command line editing can be used to edit the command history. For more information on how to use command line editing, see “Command Line Editing” on page 10.

## Example

### CODE EXAMPLE 3-26 history Command

```
schostname:SC> history  
May 07 16:29:21 : showboards  
May 07 16:29:24 : showdate  
May 07 16:29:29 : history
```

# password

Sets the password for the platform or the domain.

## Scope

platform shell, domain shell

## Syntax—Platform Shell

```
password [-d domainID]
```

```
password -h
```

## Syntax—Domain Shell

```
password
```

```
password [-h]
```

## Options/Parameters

-d *domainID* is domain a, b, c, or d.

-h displays help for this command.

## Description

Sets the password for the platform or the domain. There are separate passwords for each domain and for the platform. If you set a password, entering the password is required for access to the shell or console.

---

**Note** – If you have a redundant SC configuration and are running firmware version 5.13.0, be aware that the password on the main system controller for the platform shell is also the *same* password on the spare system controller.

---

It is very important to set the password for the platform and each domain even if a domain is not being used. This prevents people from creating and activating unauthorized domains.

Prior to allowing the password to be changed, the current password will be authenticated. Changed passwords take effect immediately. The old password will no longer be accepted.

You can remove the password by pressing Return at the `Enter new password` and `Enter new password again` prompts.

OpenBoot PROM passwords are different from the platform and domain passwords. The OpenBoot PROM in each domain supports the OpenBoot PROM security mode, which is a standard feature of the OpenBoot PROM software. For more information on the OpenBoot PROM security mode password, see your OpenBoot PROM documentation.

## See Also

“Security” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*

## Examples

You will see the following prompt (CODE EXAMPLE 3-27), when a password is already set and you type the `password` command at either the platform shell or the domain shell.

### CODE EXAMPLE 3-27 `password` Command Example With a Password Already Set

```
schostname:SC> password  
Enter current password:  
Enter new password:  
Enter new password again:  
schostname:SC>
```

If currently there is not a password assigned to the shell you are entering, you will not be prompted for the current password (CODE EXAMPLE 3-28).

### CODE EXAMPLE 3-28 `password` Command Example With No Password Set

```
schostname:SC> password  
Enter new password:  
Enter new password again:  
schostname:SC>
```

# poweroff

Powers off components.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
poweroff [-y|-n] all|grid#|component_name [component_name ... ]
```

```
poweroff -h
```

## Syntax for the Domain Shell

```
poweroff [-y|-n] all|component_name [component_name ... ]
```

```
poweroff -h
```

## Options/Parameters

-h displays help for this command.

-y will answer yes to any question. This option is potentially hazardous. You can forcefully power off a component with the -y option.

-n answers no to any question. You cannot forcefully power off a component with the -n option.

*component\_name* is the component name.

Platform shell *component\_name*(s):

- all turns off all currently controllable components.
- Power grid (grid0, grid1). The Sun Fire 6800 system has two power grids: *grid0* and *grid1*. Grid 1 controls power supplies ps3, ps4, and ps5. All other mid-range systems have one power grid, *grid0*. This grid controls power supplies ps0, ps1, and ps2.
- Power supply (ps0 - ps5)
- Spare system controller (ssc0 or ssc1)
- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)
- Repeater board (rp0 - rp3)

- Fan tray (ft0 - ft3)

Domain shell *component\_name(s)*:

- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)

---

**Note** – The specified board must be in the current domain.

---

## Description

Powers off a component or a list of components.

For the domain shell, the specified board must be in the current domain.

All slots listed by `showboards` can be powered off except for the main system controller and the ID board (ID0). A powered-off component will not be monitored. Use the `showboards` command to display the power status of each board.

You can power off any component except when the board is in the `Active` state (see the `showboards` command to display the board state). When a component is in the `Active` state, a warning is displayed on the console that tells you the entire domain will go down. You are asked to confirm with a warning of the consequences.

If a board is active in a domain and you forcefully power off a device (board), the keyswitch for the domain will be set to standby. To change the keyswitch setting, use the `setkeyswitch` command.

You can power off the spare SC by running the `poweroff` command from the main SC. When the spare SC is powered off, the hot plug LED is illuminated.

Repeater boards do not have an `Active` state. You cannot power off a Repeater board if it is being used by a domain.

You can turn off power supplies if turning off the power supply would leave sufficient power in the power grid to support the current load of the power grid. This varies with the configuration. If you attempt to power off a power supply that is required to support the current load, the power supply will not be powered off and a message stating why is displayed. You can power off redundant power supplies.

If powering off a fan would result in insufficient cooling, it cannot be powered off.

## See Also

`poweron`, `setkeyswitch`, `showboards`

## Examples

To power off CPU/Memory board, *sb2*, from the platform shell (CODE EXAMPLE 3-29), type:

**CODE EXAMPLE 3-29** `poweroff` Command Example Showing Powering Off *sb2*

```
schostname:SC> poweroff sb2
```

To power off all power supplies, fan trays, and system boards (CODE EXAMPLE 3-30), type:

**CODE EXAMPLE 3-30** `poweroff` Command Example Showing Powering Off All Power Supplies, Fan Trays, and System Boards in the Platform Shell

```
schostname:SC> poweroff all
```

---

**Note** – From the domain shell, you cannot power off power supplies, fan trays, Repeater boards, or power grids. To power off these components, connect to the platform shell.

---

# poweron

Powers on components.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
poweron all | grid# | component_name [ component_name . . . ]
```

```
poweron -h
```

## Syntax for the Domain Shell

```
poweron all | component_name [ component_name . . . ]
```

```
poweron -h
```

## Options/Parameters

-h displays help for this command.

Platform shell *component\_name*(s):

- all is all power supplies and boards.
- Power grid (*grid0*, *grid1*). The Sun Fire 6800 system has two power grids: *grid0* and *grid1*. Grid 1 controls power supplies *ps3*, *ps4*, and *ps5*. All of the other mid-range systems have one power grid, *grid0*. This grid controls power supplies *ps0*, *ps1*, and *ps2*.
- Power supply (*ps0* - *ps5*)
- Spare system controller (*ssc0* or *ssc1*)
- CPU/Memory board (*sb0* - *sb5*)
- I/O assembly (*ib6* - *ib9*)
- Repeater board (*rp0* - *rp3*)
- Fan tray (*ft0* - *ft3*)

Domain shell *component\_names*:

- all powers on all CPU/Memory boards and I/O assemblies in the domain
- CPU/Memory board (*sb0* - *sb5*)
- I/O assembly (*ib6* - *ib9*)

The specified board must be in the current domain.

## Description

Powers on a component or a list of components. You *must* specify *component\_name(s)*.

From the domain, only components that are assigned to the domain can be powered on. To power on a component, the power grid power must also be on.

Fan trays are automatically powered on when a power supply is powered on. Fan trays obtain power from the power grid(s).

If the spare system controller was powered off, you can power on the spare system controller by running the `poweron` command from the main system controller. The spare system controller will power on automatically when the system is turned on or when the spare system controller is inserted into a system that has power. When the spare system controller is powered on, the hot plug LED is not illuminated.

## See Also

`poweroff`, `setkeyswitch`, `showboards`

## Examples

To power on CPU/Memory board, `sb2` from the platform shell (CODE EXAMPLE 3-31), type:

**CODE EXAMPLE 3-31** `poweron` Example Powering On `sb2`

```
schostname:SC> poweron sb2
```



To power on CPU/Memory boards and I/O assemblies in the domain (CODE EXAMPLE 3-32), type:

**CODE EXAMPLE 3-32** `poweron` Example Showing Powering On All CPU/Memory Boards and I/O Assemblies in Domain A

```
shostname:A> poweron all
```

---

**Note** – From the domain shell, you cannot power on power supplies, Repeater boards, fan trays, or power grids. To power on these components, use the platform shell.

---

# reboot

Reboots the system controller.

## Scope

platform shell

## Syntax

```
reboot [-y|-n]
```

```
reboot -h
```

## Options/Parameters

-y does not prompt for confirmation.

-n does not execute this command. Confirmation is requested.

-h displays help for this command.

## Description

Reboots the system controller. The `reboot` command will not change any configuration settings. Active domains will continue to run.

The keyswitch may be set to off if the system controller reboots while a keyswitch operations is in progress.



---

**Caution** – Rebooting interrupts any current operation. This includes keyswitch operations, rebooting the Solaris operating environment, testing boards, SNMP, and so on.

---

## Examples

To reboot the system controller (CODE EXAMPLE 3-33), type:

### CODE EXAMPLE 3-33 `reboot` Command Example

```
schostname:SC> reboot  
Are you sure you want to reboot the System Controller now? [no]
```

## reset

Resets the domain.

## Scope

domain shell

## Syntax

```
reset [-y|-n][-x|-a]
```

```
reset -h
```

## Options/Parameters

With no arguments, typing `reset` is the same as typing `reset -x`.

`-y` answers yes to the question asked (executes the command).

`-n` answers no to the question asked (does not execute the command).

`-h` displays help for this command.

`-x` resets via XIR (externally initiated reset). Use XIR to try to obtain diagnostic data (default).

`-a` reset is equivalent to the OpenBoot PROM `reset-all` command.

## Description

This command resumes the domain console. Resetting the domain is not allowed if the keyswitch is in the secure position, nor is it possible if the keyswitch is configured in either the off or standby position. Before you can reset the domain, change the keyswitch position to on.

By default, `reset` uses XIR (externally initiated reset) to reset the CPUs in the domain. The XIR forces control of the domain into the OpenBoot PROM and begins the OpenBoot PROM error reset recovery actions. The error reset recovery actions preserve most domain states to allow collecting data needed for debugging the hardware and software, including a Solaris operating environment core file. The OpenBoot PROM error reset recovery actions are controlled by setting the OpenBoot PROM `error-reset-recovery` configuration variable. For the definition of this variable and the various settings, see TABLE 3-10 in “`setupdomain`” on page 82.

Note that the OpenBoot PROM `reset` command *does not* generate a core file as this command does.

You cannot reset a domain that has been paused. The domain is paused automatically when hardware detects an error. Messages on the domain console indicate that the domain is paused.

To take a domain out of the paused state:

1. Turn the keyswitch off with `setkeyswitch off`.
2. Turn the keyswitch on with `setkeyswitch on`.

## See Also

`resume`, `setkeyswitch`, `setupdomain`, `showdomain`, `showkeyswitch`, `showresetstate`, “Domain Not Responding” section in the “Troubleshooting” chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*. This section describes how to recover from a hung or paused domain.

## Examples

To reset the domain (from a domain shell), type the following (CODE EXAMPLE 3-34). This command uses XIR (externally initiated reset) to try to obtain diagnostic data.

**CODE EXAMPLE 3-34** `reset` Command Example From Domain A

```
schostname:A> reset
```

To perform the equivalent operation as the OpenBoot PROM `reset-all` command (CODE EXAMPLE 3-35), type:

**CODE EXAMPLE 3-35** `reset -a` Command Example From Domain A

```
schostname:A> reset -a
```

# restoreconfig

Restores the platform and domain configurations from a server.

## Scope

platform shell

## Syntax

```
restoreconfig [-y|-n] -f url
```

```
restoreconfig -h
```

## Options/Parameters

-h displays help for this command.

-y does not prompt for confirmation.

-n does not execute this command if confirmation is required.

-f specifies a URL, which must use the ftp protocol.

*url* is the directory containing the data files. The URL must point to a directory and not a file. Examples are:

```
ftp://userid:password@hostname/path
```

```
ftp://hostname/path
```

For details on ftp URLs, see “Using FTP URLs” on page 8.

---

**Note** – The hostname you enter can be the host name or the host name with a fully qualified domain name, such as *schostname* or *schostname.eng.sun.com*. You *cannot* use a partial domain name.

---

## Description

Restores the platform and domain configurations from a server, using the two data files created by the `dumpconfig` command, *schostname.nvci* and *schostname.tod*. The files are located at the specified URL, which must point to a directory and use the ftp protocol. The `restoreconfig` command prompts you for a new platform password.



---

**Caution** – The `dumpconfig` command should have been executed when you set up the system. It is also used in the procedure for replacing a failed system controller in single SC configurations. For general instructions on how to use `dumpconfig`, see “To Use `dumpconfig` to Save Platform and Domain Configurations” in the chapter “System Power On and Setup” in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

---

This command requires that all domains be powered off with the `setkeyswitch off` command. The main and spare system controllers are automatically rebooted when the configuration is restored. The date and time need to be set after the configuration has been restored and the system controller has been rebooted. You also need to activate all domains with `setkeyswitch on`, since this command restores all domains in the off keyswitch position.

---

**Note** – Check the configuration of the platform and the domains before activating any domains to ensure that the desired configuration has been restored. In the platform shell use `showplatform`, `showboards`, and `showdate`. In each domain shell use `showdomain`, `showboards`, and `showdate`.

---

## See Also

`dumpconfig`, `setkeyswitch`, `setdate`, `showboards`, `showdate`, `showdomain`, `showplatform`

## Example

CODE EXAMPLE 3-36 shows an example of the `restoreconfig` command with the required `-f` option.

### CODE EXAMPLE 3-36 `restoreconfig` Example

```
schostname:SC> restoreconfig -f ftp://hostname/path
```

```
The system controller will be rebooted when the restore is complete. The date will need to be set in the platform and each domain.
```

```
Do you want to restore the system controller configuration now [no] yes
```

```
Retrieving: ftp://hostname/path/schostname.nvci
```

```
Retrieving: ftp://hostname/path/schostname.tod
```

```
Verifying data
```

**CODE EXAMPLE 3-36** restoreconfig Example *(Continued)*

```
NVCI has been restored
TOD has been restored
```

```
Restore complete. The system controller is being rebooted.
The date will need to be set in the platform and each domain.
```

```
Software Reset . . .
```



## resume

Exits the domain shell and resumes access to the domain console.

### Scope

domain shell

### Syntax

```
resume [-h]
```

### Options/Parameters

-h displays help for this command.

### Description

Exits the domain shell and resumes access to the domain console. This command requires an active domain. If the domain is not active, there is no domain console and there is nothing to resume. When you connect to the domain console, there may not be any output displayed. If either the OpenBoot PROM or the Solaris operating environment is running in the domain, press the Return key to get the prompt. Otherwise, POST output is displayed (may be a delay in displaying POST output).

Other system controller commands that also resume the domain console are listed in the following section.

### See Also

break, reset, setkeyswitch

## Example

### CODE EXAMPLE 3-37 `resume` Command Example in Active Domain A

```
schostrname:A> resume
```

Note that no other output is seen. If the OpenBoot PROM is at the `ok` prompt or the Solaris operating environment is at the `login:` prompt, you must press the Enter key in order to see the prompt. Otherwise, you will need to wait for output to be generated by POST, the OpenBoot PROM, or the Solaris operating environment.

If the domain is hung, there will be no output. However, when a domain displays no output, this does not necessarily mean that the domain is hung.

# setdate

Sets the date and time.

## Scope

platform shell, domain shell

## Syntax

```
setdate [-v] [-t time zone] [mmdd] HHMM
```

```
setdate [-v] [-t time zone] mmddHHMM [[cc]yy][.SS]
```

```
setdate [-v] [-r datehost]
```

```
setdate [-v] -t time zone
```

```
setdate [-v] -t GMT<+/-> offset from GMT (TABLE 3-7)
```

```
setdate -h
```

## Options/Parameters

-t *time zone* sets the time zone using the time zone abbreviation (TABLE 3-7). Only non-daylight savings time zones can be specified. If you are in an area with daylight time or summer time, the time and time zone are adjusted automatically for daylight time or summer time.

-t GMT<+/->*offset* from Greenwich Mean Time—*GMT* (TABLE 3-7).

*mm* is the month number.

*dd* day is the number in the month.

*HH* is the hour number (24-hour clock).

*MM* is the minute number.

*cc* is the century minus 1.

*yy* is the last two digits of the year number.

*SS* is the second number. For the century minus 1 value, use 21 (Twenty first century) minus 1, which yields 20.

`-r datehost` sets the current time using `rdate`. The host must be a valid system name and the system must support `rdate` requests. The hostname you enter can be the host name or the host name with a fully qualified domain name, such as `schostname` or `schostname.eng.sun.com`. You *cannot* use a partial domain name.

`-v` is verbose mode. Displays detailed information about the time zone that is set.

`-h` displays help for this command.

**TABLE 3-7** Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time

Time Zone Abbreviation	Time Zone Name	Offset From Greenwich Mean Time (GMT)
ACT	Australian central time	GMT+9.5
AET	Australian eastern time	GMT+10
AGT	Argentina standard time	GMT-3
ART	Arabic (Egypt) standard time	GMT+2
AST	Alaska standard time	GMT-9
BET	Brazil eastern time	GMT-3
BST	Bangladesh standard time	GMT+6
CAT	Central African time	GMT+2
CNT	Canada Newfoundland time	GMT-3.5
CST	Central standard time	GMT-6
CTT	China Taiwan time	GMT+8
EAT	Eastern African time	GMT+3
ECT	European central time	GMT+1
EET	Eastern European time	GMT+2
EST	Eastern standard time	GMT-5
HST	Hawaii standard time	GMT-10
IET	Indiana eastern standard time	GMT-5
IST	India standard time	GMT+5.5
JST	Japan standard time	GMT+9
MET	Middle East time	GMT+3.5
MIT	Midway Islands time	GMT-11
MST	Mountain standard time	GMT-7
NET	Near East time	GMT+4

**TABLE 3-7** Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time (*Continued*)

Time Zone Abbreviation	Time Zone Name	Offset From Greenwich Mean Time (GMT)
NST	New Zealand standard time	GMT+12
PLT	Pakistan Lahore time	GMT+5
PNT	Phoenix standard time	GMT-7
PRT	Puerto Rico and U S. Virgin Islands time	GMT-4
PST	Pacific standard time	GMT-8
SST	Solomon standard time	GMT+11
UTC	Universal Time Coordinated	GMT+0
VST	Vietnam standard time	GMT+7

## Description

Sets the date and time for the platform and domains. This command, when invoked from the platform, will have no effect on the date and time in each domain and vice versa. You can set up to five different times and time zones; one time and time zone for the platform and different times and time zones for each of the four domains.

If your time zone area is using daylight or summer time, this is set automatically.

---

**Note** – You cannot set the date from the system controller in a domain while the Solaris operating environment is running. To set the date while the Solaris operating environment is running, use the Solaris operating environment `date` command.

---

After you set the date and time, you can use the SNTP server to keep the date and time synchronized. For details, see “To Set the Date and Time for the Platform” in the chapter, “System Power On and Setup” in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

## See Also

`showdate`, `setupplatform`

## Examples

When you type the `setdate` command from the platform, this sets the date and time for the platform. When you type the `setdate` command from the domain, this sets the date and time for the domain.

To set the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds, type:

### CODE EXAMPLE 3-38 `setdate` Command Example in the Platform Shell

```
schostname:SC> setdate 042018152001.10  
Thu Apr 20 18:15:10 PST 2001
```

To set the date from a date host:

### CODE EXAMPLE 3-39 `setdate -r` Command Example Setting the Date From a Date Host

```
schostname:SC> setdate -r datehost  
Thu Apr 20 18:15:10 PST 2001
```

To set the time zone to Pacific Standard Time (PST), using the offset from Greenwich mean time—GMT, and the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds (TABLE 3-7), type:

### CODE EXAMPLE 3-40 `setdate -t` Command Example Setting the Time Zone to Pacific Standard Time Using the Offset From Greenwich Mean Time

```
schostname:SC> setdate -t GMT-8 042018152001.10  
Thu Apr 20 18:15:10 PST 2001
```

To set the time zone to Eastern Standard Time (EST), using the time zone abbreviations, and the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds (TABLE 3-7), type:

### CODE EXAMPLE 3-41 `setdate -t` Command Example Setting the Date and Time Zone to Eastern Standard Time Using Time Zone Abbreviations

```
schostname:SC> setdate -t EST 042018152001.10  
Thu Apr 20 18:15:10 EST 2001
```

To set just the time zone for Japan Standard Time using offsets from Greenwich Mean Time—GMT (TABLE 3-7) and *not* the date and time, type:

**CODE EXAMPLE 3-42** `setdate -t` Command Example Setting the Time Zone to Japan Standard Time Using the Offset From Greenwich Mean Time

```
schostname:SC> setdate -t GMT+9
Thu Apr 20 18:15:10 GMP+9 2001
```

To set just the time zone for European Central Time using the time zone abbreviations (TABLE 3-7) and *not* the date and time, type:

**CODE EXAMPLE 3-43** `setdate -t` Command Example Setting the Time Zone to Eastern Central Time Using Time Zone Abbreviations

```
schostname:SC> setdate -t ECT
Thu Apr 20 18:15:10 ECT 2001
```

# setdefaults

Sets the default configuration values.



---

**Caution** – This is a destructive command. Use with caution and care.

---

## Scope

platform shell, domain shell

## Syntax

```
setdefaults [-y|-n] [-c] [-p platform]
```

```
setdefaults -h
```

## Options/Parameters—Platform Shell

-y does not prompt for confirmation. The option sets the defaults for the platform shell and each domain shell.

-n does not execute the command if confirmation is requested.

-c keeps Capacity on Demand (COD) right-to-use (RTU) license keys stored in the COD license database. This option does not preserve the number of any instant access CPUs (headroom) enabled and COD RTU licenses reserved for domains.

-h displays help for this command.

-p platform resets the defaults for the platform only.

## Options/Parameters—Domain Shell

-y does not prompt for confirmation.

-n does not execute the command if confirmation is requested.

-h displays help for this command.



## Description



---

**Caution** – This command requires that all domains are inactive (not running the OpenBoot PROM, POST, or the Solaris operating environment) and the keyswitch be set to off.

---

When you run this command from the platform shell without options, the platform and domain values are set to the default values. When you run this command from the domain shell without options, only the current domain values are reset to the default values.

If a password is set, you will need to type the password in order to set the default values. When you run this command from the platform shell, the password for the platform shell is required. When you run this command from a domain shell, the password for the domain shell is required. There are separate passwords for the platform shell and each domain shell.

In addition, this command does the following:

- Sets the system controller to the default values. This command resets only the system controller setting. It does not affect the domain's configuration in the OpenBoot PROM.

If a spare System Controller board is present, SC failover is automatically enabled. Otherwise, SC failover is disabled.

- Deletes all COD RTU licenses and resets the instant access CPU (headroom) quantity and domain COD RTU license quantity to zero (0), only if you do not specify the `-c` option.
- Erases all platform and domain configurations (platform).
- Removes passwords (domain and platform).
- Reboots both the main and spare system controllers after the defaults are set.

## Example

### CODE EXAMPLE 3-44 setdefaults Example

```
schostname:SC> setdefaults
```

```
You are about to reset all configuration data to default values.  
All domain configurations, ACLs, passwords and data buffers will  
be lost.
```

```
The system controller will be REBOOTED after the defaults are set.
```

```
Do you want to restore the default values and reboot now? [no]
```

- If you answer yes, the system controller is rebooted after the defaults are set.
- If you answer no, the default values are not restored.
- If a password is set for the platform shell, the password will be required.

# setfailover

Change the state of system controller (SC) failover.

## Scope

platform shell

## Syntax

```
setfailover [-y|-n] on|off|force
```

```
setfailover -h
```

## Options/Parameters

`-y` does not prompt for confirmation.

`-n` does not execute the command if confirmation is requested.

`on` enables failover for systems that previously had failover disabled due to a failover or an operator request.

`off` disables failover. This option prevents a failover until the failover feature is re-enabled.

`force` causes a forced failover to the spare SC.

`-h` displays help for this command.

## Description

This command enables you to control automatic or manual SC failover. Be aware that if you force a failover using this command, SC failover is disabled after the manual failover occurs. For further information on SC failover, refer to the “System Controller Failover” chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

## See Also

`setupplatform`, `showfailover`, `showlogs`, `showplatform`, `showsc`

## Examples

To manually failover from the main SC to the spare, type:

### CODE EXAMPLE 3-45 `setfailover` Command Example (run on the Spare SC) of Manual Failover

```
schostname:sc> setfailover force

SC: SSC0
Spare System Controller
SC Failover: enabled and active.
Clock failover enabled.

This will abruptly interrupt operations on the other System Controller.
This System Controller will become the main System Controller.

Do you want to continue? [no] yes
Oct 26 00:10:33 schostname Platform.SC: SC Failover: becoming main SC ...
Oct 26 00:10:37 schostname Platform.SC: Chassis is in single partition mode.
Oct 26 00:10:42 schostname Platform.SC: Main System Controller
Oct 26 00:10:42 Oct 26 00:10:42 schostname Platform.SC: Added logical IP address
xxx.xxx.xxx.xxx

Oct 26 00:10:43 schostname Platform.SC: SC Failover: disabled

schostname:SC>
```

To disable SC failover, type:

### CODE EXAMPLE 3-46 `setfailover` Command Example Disabling Failover

```
schostname:SC> setfailover off
SC Failover: disabled
```

To re-enable SC failover, type:

### CODE EXAMPLE 3-47 `setfailover` Command Example Re-enabling Failover

```
schostname:SC> setfailover on
SC Failover: enabled and active.
```

# setkeyswitch

Changes the position of the virtual keyswitch to the specified value.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
setkeyswitch [-y|-n] -d domainID off
```

```
setkeyswitch -h
```

## Syntax for the Domain Shell

```
setkeyswitch [-y|-n] [off|standby|on|diag|secure]
```

```
setkeyswitch -h
```

## Options/Parameters

-h displays help for this command.

-y does not prompt for confirmation.

-n does not execute this command is confirmation is requested.

-d *domainID* (platform shell only) is the domain ID of the domain you want power off ( a, b, c, or d).

For `setkeyswitch` parameters and descriptions, see TABLE 3-8.

**TABLE 3-8** Description of the `setkeyswitch` Transition

Value	Description
<code>off</code>	Changes the board state of all the boards belonging to a domain to the Assigned board status, and places the boards in low-power mode, which allows you to remove the boards from the system. The domain will not be initialized at system power on.
<code>standby</code>	Changes the board state of all the boards belonging to a domain to the Assigned board state and turns on the boards. The domain will not be initialized at system power on, but the boards that comprise the domain will be powered on.
<code>on</code>	Powers on and initializes the domain. The system controller brings the domain into OpenBoot PROM through POST. If the OpenBoot PROM has <code>auto-boot?</code> set to <code>true</code> , then the Solaris operating environment boots automatically.
<code>diag</code>	Similar to the <code>setkeyswitch on</code> command except the POST verbose mode and the POST diagnostic level are set to max.
<code>secure</code>	Similar to the <code>setkeyswitch on</code> command except that the <code>break</code> command and the <code>reset</code> commands are ignored. CPU/Memory board and I/O assembly flash PROM updates are not allowed.

TABLE 3-9 shows the results when you change a keyswitch setting.

**TABLE 3-9** Results of Changing the Keyswitch From the Current Setting to a New Setting

Current Setting	New Setting	Result
<code>off</code>	<code>off</code>	Keyswitch position not changed.
<code>off</code>	<code>standby</code>	Powers on all of the boards. The boards belonging to a domain should already be in the Assigned state. The domain will not be initialized at system power on.
<code>off</code>	<code>on</code>	Powers on and initializes the domain. The domain will go through POST and the OpenBoot PROM. The Solaris operating environment boots automatically unless you have set the OpenBoot PROM <code>auto-boot?</code> set to <code>false</code> .
<code>off</code>	<code>diag</code>	Similar to <code>on</code> , except POST verbose and diagnostic level is to set to maximum. The domain will boot from the <code>diag-device</code> .
<code>off</code>	<code>secure</code>	Similar to <code>on</code> , except the <code>reset</code> and <code>break</code> commands are ignored by the domain shell.

**TABLE 3-9** Results of Changing the Keyswitch From the Current Setting to a New Setting (Continued)

Current Setting	New Setting	Result
standby	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.
standby	on	Similar to setting the virtual keyswitch from off to on except that you do not have to wait for the system controller to turn on the boards and prepare them.
on	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.
on	standby	The domain becomes inactive. All of the boards remain powered on.
on	on	Keyswitch position not changed.
on	diag	Similar to the <code>setkeyswitch on</code> command except the POST verbose mode is set to on and the POST diagnostic level is set to maximum. This change is setting has no effect on a running domain.
on	secure	<code>break</code> and <code>reset</code> are ignored.
diag	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.
diag	standby	The domain becomes inactive. All of the boards remain powered on.
diag	on	Has no effect on a running domain. This operation only changes the keyswitch position and does not perform any other function since the domain is already running.
diag	diag	Keyswitch position not changed.
diag	secure	<code>break</code> and <code>reset</code> are ignored.
secure	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.
secure	standby	The domain becomes inactive. All of the boards remain powered on. The <code>break</code> and <code>reset</code> commands are no longer ignored.
secure	on	<code>break</code> and <code>reset</code> are not ignored.
secure	diag	This change is setting has no effect on a running domain.
secure	secure	Keyswitch position not changed.

## Description

Each domain has a virtual keyswitch with five positions: off, standby, on, diag, and secure. The `setkeyswitch` command changes the position of the virtual keyswitch to the specified value. The virtual keyswitch replaces the need for a physical keyswitch for each domain.

If it is not possible to access the domain shell, set the keyswitch position for the domain to off from the platform shell. Setting the keyswitch position to off for a domain without first halting the Solaris operating environment may cause problems. If the domain is running the Solaris operating environment, the `setkeyswitch` command parameters `off` and `standby` require confirmation.

If the domain is already powered on, the `setkeyswitch on`, `setkeyswitch diag`, and `setkeyswitch secure` commands will change only the position of the virtual keyswitch. If components making up a domain such as a I/O assembly or CPU/Memory board need to be powered on, the system attempts to power on these boards or devices also.

When the keyswitch position changes to `on`, `secure`, or `diag` from any keyswitch position, the domain console is resumed.

If a failure occurred during keyswitch transition, the keyswitch position is reset to `off` or `standby` after recovering from the failure.

The system controller maintains the position of each keyswitch even during power failures or physical power cycles. When you power on the system after either a power failure or system power off, the system controller activates any domains that were active when the power went off.

## See Also

`resume`, `showkeyswitch`

## Examples

To turn the virtual keyswitch on and power on the domain (CODE EXAMPLE 3-48), type:

**CODE EXAMPLE 3-48** `setkeyswitch on` Example

```
schostname:A> setkeyswitch on
```



To shut down the domain by turning the virtual keyswitch off (CODE EXAMPLE 3-49), type:

**CODE EXAMPLE 3-49** `setkeyswitch off` Example

```
schostname:A> setkeyswitch off
```

To shut down domain B from the platform shell (CODE EXAMPLE 3-50), type:

**CODE EXAMPLE 3-50** `setkeyswitch off` Example Shutting Down Domain B From the Platform Shell

```
schostname:SC> setkeyswitch -d b off
```



---

**Caution** – CODE EXAMPLE 3-50 does not properly shut down the Solaris operating environment in the domain before turning the keyswitch in domain B to the off position.

---

# setupdomain

Configures the domain.

## Scope

domain shell

## Syntax

```
setupdomain [-p part]. . .
```

```
setupdomain -h
```

## Options/Parameters

-h displays help for this command.

-p *part* configures the specified information and has the following options:

- `bootparams` configures the boot parameter information.
- `loghost` configures the log hosts.
- `snmp` configures SNMP information.

TABLE 3-10 lists the parameter values for `setupdomain`. Note that for `diag-level` definition and function of `default` and `max` are the same. Because the tests for `mem1` and `mem2` take a long time to run, use the `default` or `max` values. However, if you suspect memory problems, then use the `mem1` or `mem2` values.

**TABLE 3-10** Parameter Values for the `setupdomain` Command

Parameter	Value	Description
diag-level	init	Only system board initialization code is run. No testing is done. This is a very fast pass through POST.
	quick	All system board components are tested using few tests with few test patterns.
	default (default value)	All system board components are tested with all tests and test patterns, except for memory and Ecache modules. For memory and Ecache modules, all locations are tested with multiple patterns. More extensive, time-consuming algorithms are not run at this level. This is the same as <code>max</code> .

**TABLE 3-10** Parameter Values for the `setupdomain` Command (Continued)

Parameter	Value	Description
	max	All system board components are tested with all tests and test patterns. This is the same as default.
	mem1	Runs all tests at the default level plus more exhaustive DRAM and SRAM test algorithms. These tests take a long time to run.
	mem2	This is the same as mem1 with the addition of a DRAM test that does explicit compare operations of the DRAM data. These tests take a long time to run.
verbosity-level	off	No status messages are displayed. POST output is always displayed. However, with the value set to off, there is very little POST output displayed and there may be a long time between output displays. This depends on your hardware configuration and the setting of diag-level.
	min (default value)	Test names status messages, and error messages are displayed.
	max	Subtest trace messages are displayed.
error-level	off	No error messages are displayed.
	min	The failing test name is displayed.
	max (default value)	All relevant error status is displayed.
interleave-scope	within-cpu	The memory banks on a CPU will be interleaved with each other. Each CPU may have up to four logical memory banks.
	within-board (default value)	The memory banks on a system board will be interleaved with each other.
	across-boards	The memory will be interleaved on all memory banks across all of the boards in the domain.
		With this value, you cannot remove a CPU/Memory board from the system using DR.
interleave-mode	optimal (default value)	The memory is mixed-size interleaving in order to gain optimal performance.
	fixed	The memory is fixed-size interleaving.
	off	There is no memory interleaving.
reboot-on-error	true (default value)	The domain is rebooted when the system controller detects a hardware error. If <code>OBP.auto-boot?</code> is set to true, the Solaris operating environment is booted.

**TABLE 3-10** Parameter Values for the `setupdomain` Command (Continued)

Parameter	Value	Description
	<code>false</code>	The domain is paused when the system controller detects a hardware error. To recover, you must manually reset the domain (turn the domain off, then on by using the <code>setkeyswitch</code> command).
<code>error-policy</code>	<code>diagnose</code> (default value)	A diagnosis message, along with the reporting and logging of error messages, is displayed when a problem with a domain component is encountered.
	<code>display</code>	Error messages are reported and logged.
<code>OBP.use-nvramrc?</code>		This parameter is the same as the OpenBoot PROM <code>nvramrc?</code> parameter. This parameter uses aliases that are stored in <code>nvramrc</code> .
	<code>true</code>	The OpenBoot PROM executes the script stored in <code>nvramrc</code> if this parameter is set to <code>true</code> .
	<code>false</code> (default value)	The OpenBoot PROM does not evaluate the script stored in <code>nvramrc</code> if this parameter is set to <code>false</code> .
<code>OBP.auto-boot?</code>		Controls booting of the Solaris operating environment.
	<code>true</code> (default value)	If this value is <code>true</code> , the domain boots automatically after POST has run.
	<code>false</code>	If this parameter value is set to <code>false</code> , you will obtain the OpenBoot PROM <code>ok</code> prompt after POST runs, from which you must type a <code>boot</code> command to boot the Solaris operating environment.
<code>OBP.error-reset-recovery</code>		Controls the behavior of the domain after an externally initiated reset (XIR) as well as a red mode trap.
	<code>sync</code> (default value)	The OpenBoot PROM invokes <code>sync</code> . A core file is generated. If the invocation returns, the OpenBoot PROM performs a reboot.
	<code>none</code>	The OpenBoot PROM prints a message describing the reset trap that triggered the error reset and passes control to the OpenBoot PROM <code>ok</code> prompt. The message describing the reset trap type is platform specific.

**TABLE 3-10** Parameter Values for the `setupdomain` Command (Continued)

Parameter	Value	Description
	boot	<p>The OpenBoot PROM firmware reboots the domain. A core file is not generated. Rebooting a domain occurs using the OpenBoot PROM settings for <code>diag-device</code> or <code>boot-device</code>, depending on the value of the OpenBoot PROM configuration variable <code>diag-switch?</code></p> <p>If <code>diag-switch?</code> is set to true, the device names in <code>diag-device</code> will be the default for boot. If <code>diag-switch?</code> is set to false, the device names in <code>boot-device</code> will be the default for boot.</p>
Loghost		The name or the IP address of the loghost for this domain (SNMP).
Log Facility		The log facility is defined by the Solaris operating environment <code>syslogd</code> in <code>/etc/syslog.conf</code> . The default is <code>local0</code> .
Domain Description		Enter a brief description for this domain. For example, its function.
Domain Contact		Enter the name of the primary domain administrator.
Trap Hosts		Enter the name or the IP address of the SNMP trap host for this domain. The SNMP agent sends traps to the trap host on a SNMP default port number (162). An optional and different port number can be used other than the default port number. The format of the trap host is <code>host[:port]</code> .
Public Community String		<p>Community string for SNMP readers. The default value is A-public, B-public, and so on.</p> <p><b>Note</b>—For SNMP clients such as the Sun Management Center 3.0 software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.</p>
Private Community String		<p>Community string for readers and writers. The default value is A-private, B-private, and so on.</p> <p><b>Note</b>—For SNMP clients such as the Sun Management Center 3.0 software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.</p>

## Description

Configures domain specific values. Configures the domain. You can configure each domain differently and independently This command is an interactive command. You are prompted for parameters to configure.

## See Also

password, setdate, showdate, showdomain, and the section “To Configure Domain-Specific Parameters” in the “System Power On and Setup” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

## Example

**CODE EXAMPLE 3-51** Variables for the setupdomain Command

```
schostname:A> setupdomain

Domain Boot Parameters
-----
diag-level [default]:
verbosity-level [min]:
error-level [max]:
interleave-scope [within-board]:
interleave-mode [optimal]:
reboot-on-error [true]:
error-policy [diagnose]:
OBP.use-nvramrc? [<OBP default>]:
OBP.auto-boot? [<OBP default>]:
OBP.error-reset-recovery [<OBP default>]:

Log hosts
-----
Loghost [ ]: The hostname or the IP address of the loghost for this domain.
Log Facility [local0]: Defined by the Solaris operating environment syslogd in /etc/syslog.conf.

SNMP
-----
Domain Description[]: A brief description for this domain (its function).
Domain Contact[ ]: The name of the primary domain administrator.
Trap Hosts [ ]: The name or IP address of the SNMP trap host for this domain.
Public Community String [ ]: The community string for SNMP readers.
Private Community String [ ]: The community string for readers and writers.
```

**CODE EXAMPLE 3-51** Variables for the `setupdomain` Command (*Continued*)

```
schostname: A>
```

---

**Note** – The default value displayed by the software for the `OBP.*` parameters is `OBP default`.

---

If you enter an invalid value, the system controller displays “invalid entry” and prompts you to enter a proper value (TABLE 3-10). If a value is not specified, it retains its current value. When an invalid entry is entered, it also lists the valid responses. Typing a dash ( - ), clears the entry.

Current values are displayed in [ ]. TABLE 3-10 lists the `setupdomain` parameter values.

# setupplatform

Configures the platform specific variables.

## Scope

platform shell

## Syntax

```
setupplatform [-p part]...
```

```
setupplatform -h
```

## Options/Parameters

-h displays help for this command.

-p *part* configures the specified part and has the following options:

- `acls` is ACLs (access control list)
  - clears the boards from the ACL
  - + adds all boards to the ACL
  - a *board* adds a board
  - d *board* deletes a board
- `cod` configures the Capacity on Demand parameters and has the following options:
  - headroom-number* configures only the COD instant access CPU quantity.
  - d *domainID proc-RTUs* reserves a specific number of COD RTU licenses for a specified domain (a, b, c, or d).
- `loghost` configures loghost and facility with the default of local0. The loghost is the IP address or host name of the syslog loghost. The host name you enter must be a complete host name (not a partial one).
- `network` is the network settings.
- `partition` sets the partition mode.
- `sc` configures system controller behavior.
- `security` controls telnet access to the system controller so that telnet services are either immediately available or disabled. If you disable telnet access, any currently active telnet sessions are ended immediately, including the session used to run the command. You can also set a timeout period for idle telnet or serial port connections to the system controller.



- `snmp` is Simple Network Management Protocol (SNMP) configuration.
- `sntp` is Simple Network Time Protocol (SNTP)

## Description

Configures parameters for the platform. This command is an interactive command. You are prompted for parameters to configure. For parameters to configure, see TABLE 3-11.

**TABLE 3-11** `setupplatform` Parameter Values

Parameter	Description
Is the System Controller on a Network?	If the answer is no, leave all network parameters as unconfigured. You will be prompted to enter the hostname. If the answer is yes, you are prompted to configure the network settings.
Use DHCP or static network settings?	<ul style="list-style-type: none"> <li>• <code>DHCP</code> means the system controller network configuration is retrieved from a DHCP server.</li> <li>• <code>Static</code> means the network settings will be manually configured. If you select the static setting, you will be asked for the following network parameters.</li> </ul>
Hostname (Network parameter)	The human readable network identity for this system controller.
IP address (Network parameter)	The network identity used by computers.
Netmask (Network parameter)	For this value, specify how much of the address should be reserved for subdividing networks into subnetworks. Must be specified in dot-notation address. For example 255.255.255.0.
Gateway (Network parameter)	IP address of the gateway.
DNS Domain (Network parameter)	Domain name. For example, <code>xxx.xxx.com</code> .
Primary DNS Server (Network parameter)	IP address of your primary DNS server.
Secondary DNS Server (Network parameter)	IP address of your secondary DNS server.
Loghost	The name or the IP address of the loghost for this domain (SNMP).
Log Facility	The log facility. Facility is defined by the Solaris operating environment <code>syslogd</code> in <code>/etc/syslog.conf</code> . The default is <code>local0</code> .

**TABLE 3-11** setupplatform Parameter Values (Continued)

Parameter	Description
SNTP server	The name of the Simple Network Time Protocol (SNTP) server that automatically tracks and corrects the drift (difference) between the local system clocks managed by each system controller.
Platform Description	Default value is the platform model name.
Platform Contact	The name of the person who is responsible for this system. This name will be used by SNMP. Note that if SNMP is not going to be enabled, this field is informational only.
Platform Location	Location of the system (such as a room and/or the location inside of the room). This information will be used by SNMP. Note that if SNMP is not going to be enabled, this field is informational only.
Enable SNMP Agent?	If you answer yes, the Simple Network Management Protocol (SNMP) agent enables or disables SNMP. The default is SNMP disabled.
Trap Hosts	IP address or name of the SNMP trap hosts. The SNMP agent sends traps to the trap host on an SNMP default port number (162). An optional and different port number can be used other than the default port number. The format of the trap host is host[:port].
Public Community String	Community string for SNMP readers. The default is P-public.  <b>Note</b> —For SNMP clients such as the Sun Management Center 3.0 software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.
Private Community String	Community string for readers and writers. The default is P-private.  <b>Note</b> —For SNMP clients such as the Sun Management Center 3.0 software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.

**TABLE 3-11** setupplatform Parameter Values (Continued)

Parameter	Description
ACL for Domain A	<p>Access control list (ACL). If the board name is listed in the ACL for the specific domain, then you have permission to perform <code>addboard</code> or <code>deleteboard</code> requests on that board. Default value is to have all slots assigned to all domains.</p> <ul style="list-style-type: none"> <li>• Typing the prefix <code>-d</code> deletes the slot from ACLs of this domain.</li> <li>• Typing a prefix of <code>-a</code> (add) adds a slot.</li> <li>• Typing a dash ( <code>-</code> ) clears the entire list.</li> <li>• Typing a <code>+</code> (plus sign) adds all.</li> <li>• Pressing the Return key does not change the value displayed inside <code>[ ]</code>.</li> </ul>
ACL for Domain B	See the description for ACL for Domain A.
ACL for Domain C	See the description for ACL for Domain A.
ACL for Domain D	See the description for ACL for Domain A.
PROC RTUs installed	Displays the number of COD RTU licenses currently installed
PROC Headroom Quantity (0 to disable, 4 MAX) [x]	<p>The number of COD instant access CPUs to be enabled.</p> <ul style="list-style-type: none"> <li>• The maximum number of instant access CPUs that can be enabled (4) is displayed inside the parenthesis.</li> <li>• Specify 0 to disable the instant access CPU quantity only if there are no instant access CPUs currently in use.</li> <li>• The current number of instant access CPUs enabled is displayed inside the brackets.</li> </ul>
PROC RTUs reserved for domain A (x MAX) [y]:	<p>The number of COD right-to-use (RTU) licenses to be reserved for the domain.</p> <ul style="list-style-type: none"> <li>• The maximum number of COD RTU licenses that can be reserved for the domain is displayed inside the parenthesis.</li> <li>• The current number of COD RTU licenses allocated to the domain is displayed inside the brackets.</li> </ul>
PROC RTUs reserved for domain B (x MAX) [y]:	See the description for PROC RTUs reserved for domain A.
PROC RTUs reserved for domain C (x MAX) [y]:	See the description for PROC RTUs reserved for domain A.
PROC RTUs reserved for domain D (x MAX) [y]:	See the description for PROC RTUs reserved for domain A.
SC POST diag level	The system controller boot parameters (diag level) can be set to: <code>off</code> , <code>min</code> , or <code>max</code> .

**TABLE 3-11** setupplatform Parameter Values (Continued)

Parameter	Description
off	Only system board initialization code is run. No testing is done. POST is not run.
min	All System Controller board components are tested using few locations with few test patterns.
max	All System Controller board components are tested with all tests and test patterns.
Enable SC failover?	If the answer is <i>yes</i> and you have a redundant SC configuration, SC failover is enabled (activated). If the answer is <i>no</i> , SC failover is disabled until you re-enable failover.
Logical Hostname or IP Address	Name of the logical host or IP address that always identifies the working main SC. When an SC failover occurs, this logical hostname or IP address is associated (floats) with the new main SC.
Enable telnet servers?	<p>The floating IP address must be different from the IP address used to identify the system controller.</p> <p>If the answer is <i>yes</i>, telnet service to the system controller is available immediately.</p> <p>If the answer is <i>no</i>, you are asked to confirm the termination of active telnet sessions. After the confirmation, all active telnet sessions are ended immediately.</p>
Idle connection timeout (in minutes; 0 means no timeout)	<p>The timeout period, in minutes, for idle telnet and serial port connections. A 0 (zero) indicates there is no timeout period. If you specify a timeout period, the actual timeout will occur within one minute after the specified timeout period. Use the <code>connections</code> command to verify the telnet and serial port connections.</p> <p><b>Note</b>— The domain boot process, which does not require user input, is considered to be idle. If your timeout period is shorter than the domain boot time, connections can be dropped during domain bootup. If the connection is dropped, the boot process continues to completion, but the console will not be available until the boot process reaches the OBP state.</p>
Configure chassis for single or dual partition mode?	The two options are single or dual. The default is single. For a description of single and dual partition mode, refer to the “Overview” chapter in the <i>Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual</i> .

## See Also

`password`, `setdate`, `setupdomain`, `showdate`, `showdomain`, `showplatform`, and the section “Setting Up the Platform” in the “System Power On and Setup” chapter of the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*. This chapter provides a step-by-step procedure on setting up the platform using `setupplatform`.

## Example

### CODE EXAMPLE 3-52 setupplatform Output

```
schostname:SC> setupplatform

Network Configuration
-----
Is the system controller on a network? [yes]:
Use DHCP or static network settings? [DHCP]:
Network settings? [static]:
Hostname []:
IP Address []:
Netmask []:
Gateway []:
DNS Domain []:
Primary DNS Server []:
Secondary DNS Server []:

Loghosts
-----
Loghost [ ]:
Log Facility [local0]:

SNTP
----
SNTP server [ntp1]:

SNMP
----
Platform Description [Sun Fire 6800]: System type
Platform Contact [ ]: Platform administrator's name
Platform Location [ ]: Geographic system location
Enable SNMP Agent? [yes]:
Trap Hosts []: SNMP trap host IP address or name
Public Community String [P-public]: Community string for SNMP readers
Private Community String [P-private]: Community string for readers and writers

ACLs
----
ACL for domain A [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain B [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain C [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain D [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
```

**CODE EXAMPLE 3-52** setupplatform Output (Continued)

```
COD
---
PROC RTUs installed: 8
PROC Headroom Quantity (0 to disable, 4 MAX) [0]: Number of instant access CPUs to be enabled
PROC RTUs reserved for domain A (6 MAX) [0]: Number of domain COD RTU licenses reserved
PROC RTUs reserved for domain B (6 MAX) [0]: Number of domain COD RTU licenses reserved
PROC RTUs reserved for domain C (4 MAX) [0]: Number of domain COD RTU licenses reserved
PROC RTUs reserved for domain D (4 MAX) [0]: Number of domain COD RTU licenses reserved

SC
--
SC POST diag Level [min]:
Enable SC failover? [yes]:
Logical Hostname or IP address [spl-sc]:

Security Options
-----
Enable telnet servers? [yes]:no
This will disconnect all active telnet sessions. Are you sure? [no] yes
Idle connection timeout (in minutes; 0 means no timeout) [0]: 3

Partition Mode
-----
Configure chassis for single or dual partition mode? [single]:

schostname: SC>
```

# showboards

Displays the assignment information and status for all of components in the system.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
showboards [-ev] [-y | -n] [-d domainID] [-p part]
```

```
showboards -h
```

## Syntax for the Domain Shell

```
showboards [-aev] [-y | -n] [-p part]
```

```
showboards -h
```

## Options/Parameters

- a include available boards (domain only).
- e includes empty slots in the output.
- v verbose mode.
- d *domainID* specify a domain to display (platform only).
- p *part* shows only a specific part and can be:
  - board shows the board status.
  - clock shows the system clock status.
  - cpu shows CPU type, speed, and Ecache size.
  - io shows I/O information.
  - memory shows memory information for each board.
  - power shows grid information.
  - version shows version information.
- y answers yes to all questions.
- n answers no to all questions.
- h displays help for this command.



## Description

Displays the assignment information and status for all of the components in the system. For example, CPU/Memory boards, I/O assemblies, fan trays, and so on. For the domain shell, the display shows components assigned to the domain and can also include boards that are included in the ACL, and not assigned to any other domain.

## See Also

`addboard`, `deleteboard`

## Examples—Platform Shell

**CODE EXAMPLE 3-53** `showboards` Command for the Platform Shell

```
schostname:SC> showboards
```

Slot	Pwr	Component Type	State	Status	Domain
----	---	-----	-----	-----	-----
SSC1	On	System Controller	Main	Passed	-
SSC0	On	Empty Slot	Spare	-	-
ID0	On	Sun Fire 3800 Centerplane	-	OK	-
PS0	On	A145 Power Supply	-	OK	-
PS1	On	A145 Power Supply	-	OK	-
PS2	On	A145 Power Supply	-	OK	-
FT0	On	Fan Tray	Low Speed	OK	-
FT1	On	Fan Tray	Low Speed	OK	-
FT2	On	Fan Tray	Low Speed	OK	-
FT3	On	Fan Tray	Low Speed	OK	-
RP0	On	Repeater Board (F3800)	-	OK	-
RP2	On	Repeater Board (F3800)	-	OK	-
/N0/SB0	On	CPU Board	Active	Passed	A
/N0/SB2	Off	CPU Board	Assigned	Not tested	C
/N0/IB6	On	CPCI I/O board (F3800)	Active	Passed	A
/N0/IB8	Off	CPCI I/O board (F3800)	Assigned	Not tested	C

The `showboards` command with the `-v` option displays the verbose option.

The `showboards` command with the `-p` version option displays firmware version and compatibility information.

**CODE EXAMPLE 3-54** `showboards` Command with the `-p` version Option

```

schostrname:SC> showboards -p version

Component      Compatible Version
-----
SSC0           Reference 5.13.0 Build 10
/N0/IB6        Yes      5.13.0 Build 9
/N0/SB2        Yes      5.13.0 Build 9
/N0/IB8        Yes      5.12.6
/N0/SB4        Yes      5.12.6

schostrname:SC>

```

**TABLE 3-12** Output Header Definitions for the `showboards` and the `showboards -v` Command

Header	Description
Slot	Slot designator. The Nx in the slot descriptor is the node number.
Pwr	Indicates if the power status of the device is off or on.
Component type	Component description, such as System Controller, CPU Board, COD CPU Board, Fan Tray, and so on.
State	Describes board state. Possible values are: Active, Assigned, Available, and - (dash). The - board state means that the board state does not apply to this slot. For system controllers, possible values are: Main or Spare. For more information on board states, see “Board States for CPU/Memory Boards and I/O Assemblies” on page 6.
Status	Current board status. For more information on board status, see “Board Test Status” on page 6.
Domain	Indicates which domain the board belongs to. For more information on domains, refer to the “Overview” chapter of the <i>Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual</i> .
<b>Memory Information</b>	
Component	Component, such as a board, CPU, or memory DIMM.
J-No	Component label on the board.
Size	Lists the size of the DIMM.
Reason	Explains why the value is not reported.

**TABLE 3-12** Output Header Definitions for the `showboards` and the `showboards -v` Command (Continued)

Header	Description
<b>Firmware Versions</b>	
Component	Board that contains a flash PROM.
Segment	Software module: ScApp, iPOST, POST, or OBP. The value Ver indicates that all software modules have been joined to form a particular image.
Compatible	Indicates whether the given component is compatible with the <code>scapp</code> version listed in the Version column. Possible values are Yes, No, and - (dash). The dash - indicates that firmware compatibility does not apply to the given component. For system controllers, the value Reference indicates that the firmware version listed in the Version column is the baseline used to determine firmware compatibility.
In	The number that identifies the software modules for the firmware. This number is used to determine the compatibility with the firmware.
Date	The month, day, and year that the flash PROM image was created by Sun Microsystems.
Time	The time of day (hour:minutes) that the flash PROM image was created by Sun Microsystems.
Build	A number that identifies the internal build. Used by Sun Microsystems only.
Version	The firmware version compatible with the component, except for System Controllers and boards that are not powered on (No board power).
<b>Clock Status</b>	
Component	Board.
SSCx Signal	Lists the status of the SSC0 or SSC1 clock on the board.
Signal Used	Defines which clock is used by the board.
Failover	Describes if the board will respond to clock failover.
<b>I/O Status</b>	
Slot	I/O slot.
Populated	Defines if a card is installed in the slot.
Slot Description	Describes the slot. The card type cannot be determined by the system controller. The card is determined by the Solaris operating environment.

**TABLE 3-12** Output Header Definitions for the showboards and the showboards -v Command *(Continued)*

Header	Description
<b>CPU Information</b>	
Component	CPU.
Cpu Mask	Lists the CPU version.
Description	Describes the CPU type, speed, and Ecache size.
<b>Power Grids</b>	
Component	Board.
Pwr	Describes if the board is on or off.
Grid	Describes which power grid the board is in.

## Example—Domain Shell

**CODE EXAMPLE 3-55** `showboards -a` Command for the Domain Shell

```
schostname:A> showboards -a
```

Slot	Pwr	Component	Type	State	Status	Domain
----	---	-----	-----	-----	-----	-----
/N0/SB2	On	CPU	Board	Assigned	Not tested	A
SB4	On	CPU	Board	Available	Not tested	Isolated
/N0/IB6	On	CPCI	I/O Board	Assigned	Not tested	A
IB8	On	CPCI	I/O board	Available	Not tested	Isolated

# showcodlicense

Displays the current Capacity on Demand (COD) right-to-use (RTU) licenses stored in the COD license database.

## Scope

platform shell

## Syntax for the Platform Shell

```
showcodlicense [-r] [-v]
```

```
showcodlicense -h
```

## Options/Parameters

-h displays help for this command.

-r displays the license information in the raw license key format, as stored in the COD license database.

-v verbose mode. Displays both the formatted license information and raw license key format.

## Description

TABLE 3-13 describes the default COD information displayed by the showcodlicense command.

**TABLE 3-13** COD License Information

Item	Description
Description	Type of resource (processor)
Ver	Version number of the license
Expiration	None. Not supported (no expiration date)
Count	Number of RTU licenses granted for the given resource
Status	One of the following states: <ul style="list-style-type: none"><li>• GOOD – Indicates the resource license is valid</li><li>• EXPIRED – Indicates the resource license is no longer valid</li></ul>

## See Also

addcodlicense, deletecodlicense, showcodusage

## Examples

To show formatted COD RTU license key data, type:

**CODE EXAMPLE 3-56** showcodlicense Example Output – Formatted COD RTU License Data

```
schostname:SC> showcodlicense
Description Ver Expiration Count Status
-----
PROC 01 NONE 8 GOOD
```

To show COD RTU license keys, type:

**CODE EXAMPLE 3-57** showcodlicense -r Example Output – COD RTU License Keys

```
schostname:SC> showcodlicense -r
01:80d8a9ed:45135285:0201000000:8:00000000:J0WKZ8vpZ5kOgSJ7e8ZasA
```

To show COD RTU license key data in both formatted and raw format, type:

**CODE EXAMPLE 3-58** showcodlicense -v Example Output – Formatted and Raw COD RTU License Data

```
schostname:SC> showcodlicense -v
Description Ver Expiration Count Status
-----
PROC 01 NONE 8 GOOD
01:80d8a9ed:45135285:0201000000:8:00000000:J0WKZ8vpZ5kOgSJ7e8ZasA
```

# showcodusage

Displays the current usage statistics for Capacity on Demand (COD) resources.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
showcodusage [-v] [-p resource|domains|all] ...
```

```
showcodusage -h
```

## Options/Parameters

- h displays help for this command.
- p resource displays COD usage information according to resource type.
- p domains displays COD usage information according to domain type.
- p all displays COD usage information both by resource type and by domain.
- v is verbose mode.

## Description

This command shows current information on the COD RTU licenses in use. By default, the command displays a summary of COD RTU licenses used and installed, along with the current state of each resource.

## See Also

showcodlicense



## Example

To show information by resource, type:

**CODE EXAMPLE 3-59** showcodusage Example Output by Resource

```
schostname:SC> showcodusage -p resource
Resource      In Use  Installed  Licensed  Status
-----
PROC          0        4         8  OK: 8 PROCs available Headroom: 2
```

TABLE 3-14 describes the resource information displayed:

**TABLE 3-14** showcodusage Resource Information

Item	Description
Resource	The COD resource (processor)
In Use	The number of COD CPUs currently used in the system
Installed	The number of COD CPUs installed in the system
Licensed	The number of COD RTU licenses installed
Status	One of the following COD states: <ul style="list-style-type: none"><li>• OK – Indicates there are sufficient licenses for the COD CPUs in use and specifies the number of remaining COD resources available and the number of any instant access CPUs (headroom) available</li><li>• HEADROOM – The number of instant access CPUs in use</li><li>• VIOLATION – Indicates a license violation exists. Specifies the number of COD CPUs in use that exceeds the number of COD RTU licenses available. This situation can occur when you force the deletion of a COD license key from the COD license database, but the COD CPU associated with that license key is still in use.</li></ul>

To show information by domain, type:

**CODE EXAMPLE 3-60** showcodusage Example Output by Domain

```

schostrname:SC> showcodusage -p domains -v
Domain/Resource  In Use  Installed  Reserved  Status
-----
A - PROC        6       8         4
  SB0 - PROC    4       4
  /NO/SB0/P0                    Licensed
  /NO/SB0/P1                    Licensed
  /NO/SB0/P2                    Licensed
  /NO/SB0/P3                    Licensed
  SB1 - PROC    2       4
  /NO/SB1/P0                    Licensed
  /NO/SB1/P1                    Licensed
  /NO/SB1/P2                    Unlicensed
  /NO/SB1/P3                    Unlicensed
B - PROC        0       0         0
C - PROC        0       0         0
D - PROC        0       0         0
Unused - PROC   0       4         0
  SB2 - PROC    0       4
  SB2/P0                    Unused
  SB2/P1                    Unused
  SB2/P2                    Unused
  SB2/P3                    Unused

```

TABLE 3-15 describes the domain information displayed.

**TABLE 3-15** showcodusage Domain Information

Item	Description
Domain/Resource	The COD resource (processor) for each domain. An unused processor is a COD CPU that has not yet been assigned to a domain.
In Use	The number of COD CPUs currently used in the domain
Installed	The number of COD CPUs installed in the domain
Reserved	The number of COD RTU licenses allocated to the domain
Status	One of the following CPU states: <ul style="list-style-type: none"> <li>• Licensed – The COD CPU has an RTU license.</li> <li>• Unused – The COD CPU is not in use.</li> <li>• Unlicensed – A COD RTU license could not be obtained for the COD CPU, so the COD CPU cannot be used.</li> </ul>

To show information by both resource type and domain, type:

**CODE EXAMPLE 3-61** showcodusage Example Output by Both Resource and Domain

```
schostname:SC> showcodusage -v
Resource      In Use  Installed  Licensed  Status
-----
PROC          0        4         8  OK: 8 PROCs available Headroom: 2
Domain/Resource  In Use  Installed  Reserved  Status
-----
A - PROC          0         0         4
B - PROC          0         0         4
C - PROC          0         0         0
D - PROC          0         0         0
Unused - PROC     0         4         0
  SB4 - PROC      0         4
  SB4/P0
  SB4/P1
  SB4/P2
  SB4/P3
                                     Unused
                                     Unused
                                     Unused
                                     Unused
```

# showcomponent

Shows the state of a component.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
showcomponent [-v] [component] ...
```

```
showcomponent [-v] [-d domainID] ...
```

```
showcomponent -h
```

## Syntax for the Domain Shell

```
showcomponent [-v] [component] ...
```

```
showcomponent -h
```

## Options/Parameters

-h displays help for this command. Includes *component\_name* syntax.

*boardname* for CPU/Memory boards is SB0 - SB5 and for I/O assemblies is IB6 - IB9.

-d *domainID* is a, b, c, or d. Shows components assigned to the domain (platform only).

## Description

Without options, this command shows the state of all components on a board. With one or more options described in the “Syntax” section, the output shows the state of one or more components on a board.

Displays the components that have been blacklisted by the `disablecomponent` command or components that have been marked as failed when testing using the `testboard` command. This command also does the following:

- For CPU/Memory boards, provides information about CPUs, memory modules (DIMMs), and Ecache.

- Displays logical memory bank size.
- For Capacity on Demand (COD) CPU components that are not allocated a COD right-to-use (RTU) license, provides the status of such components as Cod-dis (COD disabled). For details, see the “Capacity on Demand” chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*.

## See Also

`enablecomponent`, `disablecomponent`, and the “Disable Component” section of the “Troubleshooting” chapter in the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual* for a step-by-step procedure on displaying a component.

## Examples

CODE EXAMPLE 3-62 shows sample output for the `showcomponent sb4` command. The abbreviations for the Component field are:

- Nx Node name
- SBx CPU/Memory board, where x is 0 – 5
- Px Port, where x is 0 – 3
- Bx Physical memory bank, where x is 0 – 1
- Lx Logical memory bank, where x is 0 – 3. There are two DIMMs per logical memory bank.

**CODE EXAMPLE 3-62** `showcomponent sb4` Sample Output

```

schostname:SC> showcomponent sb4

```

Component	Status	Pending	POST	Description
/NO/SB4/P0	enabled	-	pass	UltraSPARC III+, 750Mhz, 8M ECache
/NO/SB4/P1	enabled	-	pass	UltraSPARC III+, 750Mhz, 8M ECache
/NO/SB4/P2	enabled	-	pass	UltraSPARC III+, 750Mhz, 8M ECache
/NO/SB4/P3	enabled	-	pass	UltraSPARC III+, 750Mhz, 8M ECache
/NO/SB4/P0/B0/L0	enabled	-	pass	256M DRAM
/NO/SB4/P0/B0/L2	enabled	-	pass	256M DRAM
/NO/SB4/P0/B1/L1	enabled	-	pass	256M DRAM
/NO/SB4/P0/B1/L3	enabled	-	pass	256M DRAM
/NO/SB4/P1/B0/L0	enabled	-	pass	256M DRAM
/NO/SB4/P1/B0/L2	enabled	-	pass	256M DRAM
/NO/SB4/P1/B1/L1	enabled	-	pass	256M DRAM
/NO/SB4/P1/B1/L3	enabled	-	pass	256M DRAM

**CODE EXAMPLE 3-62** showcomponent sb4 Sample Output (Continued)

/NO/SB4/P2/B0/L0	enabled	-	pass	256M DRAM
/NO/SB4/P2/B0/L2	enabled	-	pass	256M DRAM
/NO/SB4/P2/B1/L1	enabled	-	pass	256M DRAM
/NO/SB4/P2/B1/L3	enabled	-	pass	256M DRAM
/NO/SB4/P3/B0/L0	enabled	-	pass	256M DRAM
/NO/SB4/P3/B0/L2	enabled	-	pass	256M DRAM
/NO/SB4/P3/B1/L1	enabled	-	pass	256M DRAM
/NO/SB4/P3/B1/L3	enabled	-	pass	256M DRAM

Definitions of the headings in CODE EXAMPLE 3-62 and CODE EXAMPLE 3-63 follow:

- Component is the name of the board and its components.
- Status is the status of the board or component.
- Pending means that the component will change to the indicated status (enabled or disabled) at the next reboot.
- POST is the status of POST (passed, failed, or untested).
- Description is a description of the board or component.

CODE EXAMPLE 3-63 shows sample output for the showcomponent ib6 command. The abbreviations for the Component field are:

- NX Node name
- IBx I/O assembly, where x is 6 - 9
- Px Port, where x is 0 - 1
- Bx Bus, where x is 0 - 1
- Cx Card in the I/O assembly, where x is 0 - 7.

**CODE EXAMPLE 3-63** showcomponent ib6 Sample Output

```
schostname:SC> showcomponent ib6
```

Component	Status	Pending	POST	Description
/NO/IB6/P0	enabled	-	pass	IO controller 0
/NO/IB6/P1	enabled	-	pass	IO controller 1
/NO/IB6/P0/B1	enabled	-	untest	66/33MHz PCI Bus
/NO/IB6/P0/B0	enabled	-	untest	33MHz PCI Bus
/NO/IB6/P0/B1/C0	enabled	-	untest	33MHz 5V Short PCI card
/NO/IB6/P0/B1/C1	enabled	-	untest	33MHz 5V Short PCI card
/NO/IB6/P0/B1/C2	enabled	-	untest	33MHz 5V Long/Short PCI card
/NO/IB6/P0/B1/C3	enabled	-	untest	33MHz 5V Long/Short PCI card

# showdate

Displays the date and time.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
showdate [-tv]
```

```
showdate -h
```

```
showdate [-v] [-d domainID]
```

## Syntax for the Domain Shell

```
showdate [-tv]
```

```
showdate -h
```

## Options/Parameters

-h displays help for this command.

-t lists available time zones.

-v is verbose mode.

-d *domainID* is a, b, c, or d. Shows the date for the domain from the platform.

## Description

Displays the current date and time for the platform and for each domain. You can obtain the date for the platform shell and each domain shell from the platform shell. If you type the command from one of the domain shells, the date and time for the current domain is returned.

## See Also

setdate

## Example—Platform Shell

**CODE EXAMPLE 3-64** showdate Command for the Platform Shell

```
schostname:SC> showdate  
Mon Apr 03 12:31:40 EDT 2001
```



# showdomain

Displays the configuration and status of the domain.

## Scope

domain shell

## Syntax

```
showdomain [-v] [-p part]. . .
```

```
showdomain -h
```

## Options/Parameters

`-v` is verbose mode.

`-p part` is a parameter for the information to display:

- `acls` shows the access control list
- `bootparams` shows boot parameter information
- `cod` shows the number of COD RTU licenses reserved for the domain
- `loghosts` shows the log hosts
- `mac` shows MAC addresses
- `snmp` shows SNMP configuration
- `status` shows domain status

`-h` displays help for this command.

## Description

Displays the configuration and status of the domain. Some of the information is not configurable, such as the MAC address and hostID, or is configurable in the platform, such as the access control list (ACL), SNMP enabled, and COD parameters. This command without any arguments displays all the parameter values configured by the `setupdomain` command.

## See Also

`setupdomain`, `showdate`, `setupplatform`, `showplatform`

## Examples

To display parameters assigned by the `setupdomain` command, type `showdomain` (CODE EXAMPLE 3-65).

### CODE EXAMPLE 3-65 `showdomain` Command Example

```
schostname:A> showdomain

Domain   Solaris Nodename   Domain Status   Keyswitch
-----   -
A        -                   Powered Off     off

diag-level = default
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
error-policy = diagnose
OBP.use-nvramrc? = <OBP default>
OBP.auto-boot? = <OBP default>
OBP.error-reset-recovery = <OBP default>

Loghost for Domain A:
Log Facility for Domain A: local0

SNMP Agent: enabled
Domain Description:
Domain Contact:
Trap Hosts:
Public Community String:
Private Community String:

ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

PROC RTUs reserved for domain A: 0

schostname:A>
```

To display the verbose output of parameters assigned by the `setupdomain` command, type `showdomain -v` (CODE EXAMPLE 3-66).

**CODE EXAMPLE 3-66** `showdomain -v` Command Example

```
schostname:A> showdomain -v

Domain   Solaris Nodename   Domain Status   Keyswitch
-----   -
A        -                  Powered Off    off

Domain Boot Parameters
-----
diag-level = default
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
error-policy = diagnose
OBP.use-nvramrc? = <OBP default>
OBP.auto-boot? = <OBP default>
OBP.error-reset-recovery = <OBP default>

                MAC Address           HostID
                -----
Domain A        XX:XX:XX:XX:XX:XX   XXXXXXXX

Loghosts
-----
Loghost for Domain A:
Log Facility for Domain A: local0

SNMP
----
SNMP Agent: enabled
Domain Description:
Domain Contact:
Trap Hosts:
Public Community String:
Private Community String:

SNMP packets received: 0
SNMP packets sent: 0
SNMP traps sent: 0
```

**CODE EXAMPLE 3-66** `showdomain -v` Command Example (Continued)

```
ACLS
----
ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

COD
---
PROC RTUs reserved for domain A: 0

schostname:A>
```

To display boot parameter information (CODE EXAMPLE 3-67), type:

**CODE EXAMPLE 3-67** `showdomain -p bootparams` Example Displaying Boot Parameter Information

```
schostname:A> showdomain -p bootparams

diag-level = default
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
error-policy = diagnose
OBP.use-nvramrc? =<OBP default>
OBP.auto-boot? = <OBP default>
OBP.error-reset-recovery = <OBP default>

schostname:A>
```

# showenvironment

Displays the current environmental status, temperatures, currents, voltages, and fan status, for the system (platform shell) or the current domain (domain shell).

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
showenvironment [-ltvuw] [-d domainID] [-p part]  
showenvironment [-ltvuw] board_name  
showenvironment -h
```

## Syntax for the Domain Shell

```
showenvironment [-ltvuw] [-p part]  
showenvironment [-ltvuw] board_name [board_name ... ]  
showenvironment -h
```

## Options/Parameters

*board\_name* is the name of the board. Displays information for this board. You can type multiple board names.

-d *domainID* is a, b, c, or d. Displays information for boards assigned to that domain.

-h displays help for this command.

-l displays the limits that apply to each selected measurement. These values are the threshold for each measurement. Exceeding the threshold causes the status to display *Max* or *Min*.

-p *part* shows a specific part (TABLE 3-16):

**TABLE 3-16** Parts for `showenvironment -p part`

Report	Description
currents	Displays currents (power supplies only)
fans	Displays fan states.
faults	Displays values that are suspected to be invalid.
temps	Displays temperatures only.
voltage	Displays voltages only.

-t prints header titles.

-u updates data now. Polls all sensors for new values.

-v is verbose mode. This is the equivalent of the options -l and -w.

-w shows warning thresholds.

## Description

Displays the current environmental status, temperatures, currents, voltages, and fan status for the system (platform shell) or the current domain (domain shell). In a domain shell, the domain can only display boards that have been assigned to the domain, such as CPU/Memory boards and I/O assemblies. The board(s) must be powered on. Boards that are not powered on are not monitored and have no environmental data.

This command displays the minimum and maximum values. If these values are exceeded, the component (and the domain using the component) will be shut down. It also shows the warning levels when the system controllers warns you that you are approaching the limits.

TABLE 3-17 describes the `showenvironment` output headers and values.

**TABLE 3-17** `showenvironment` Output Header Description

Header	Value	Description
Slot		Slot ID
Device		Device being monitored by the sensor.
Sensor		Component that measures the environmental data of the device.

**TABLE 3-17** showenvironment Output Header Description (Continued)

Header	Value	Description
Value		Value returned by the sensor.
Units		Applicable unit for the sensor.
	Degrees C	Celsius
	Volts DC	Volts
	Amps	Amps
Age		Age in seconds of the reading being displayed.
Status		Values for Status. See the Value column. When the values listed in the Value column are exceeded, a warning message will be logged.
	WARNING LOW	
	NOTICE Low	
	OK	
	NOTICE High	
	WARNING HIGH	
	error	
	failed	
	ignored	
	unavailable	
LoWarn		Displayed only with the <code>-v</code> and <code>-w</code> options. Lists values that when exceeded a warning message will be logged.
HiWarn		Displayed only with the <code>-v</code> and <code>-w</code> options. Lists values that when exceeded a warning message will be logged
Min		Displayed with the <code>-v</code> and <code>-l</code> options. This is not a warning message. Min is the lowest value when the component will be powered off (including any domains using it) to protect it.
Max		Displayed with the <code>-v</code> and <code>-l</code> options. This is not a fault condition. Max is the highest value when the component will be powered off (including any domains using it) to protect it.

## Example

The `showenvironment` command display for the platform shell returns information on the slot number, device, sensor, value, units, age, and status, as shown in CODE EXAMPLE 3-68.

**CODE EXAMPLE 3-68** `showenvironment` Platform Shell and Domain Shell Example

```
md6-sc0:SC> showenvironment
```

Slot	Device	Sensor	Value	Units	Age	Status
----	-----	-----	-----	-----	---	-----
SSC0	SBBC 0	Temp. 0	38	Degrees C	2 sec	OK
SSC0	CBH 0	Temp. 0	48	Degrees C	2 sec	OK
SSC0	Board 0	Temp. 0	27	Degrees C	2 sec	OK
SSC0	Board 0	Temp. 1	27	Degrees C	2 sec	OK
SSC0	Board 0	Temp. 2	27	Degrees C	2 sec	OK
SSC0	Board 0	1.5 VDC 0	1.49	Volts DC	2 sec	OK
SSC0	Board 0	3.3 VDC 0	3.35	Volts DC	2 sec	OK
SSC0	Board 0	5 VDC 0	5.01	Volts DC	2 sec	OK
RP0	Board 0	1.5 VDC 0	1.49	Volts DC	5 sec	OK
RP0	Board 0	3.3 VDC 0	3.37	3Volts DC	5 sec	OK
RP0	Board 0	Temp. 0	23	Degrees C	5 sec	OK
RP0	Board 0	Temp. 1	22	Degrees C	5 sec	OK
RP0	SDC 0	Temp. 0	56	Degrees C	5 sec	OK
RP0	AR 0	Temp. 0	49	Degrees C	5 sec	OK
RP2	Board 0	1.5 VDC 0	1.49	Volts DC	5 sec	OK
RP2	Board 0	3.3 VDC 0	3.37	Volts DC	5 sec	OK
RP2	Board 0	Temp. 0	22	Degrees C	5 sec	OK
RP2	Board 0	Temp. 1	23	Degrees C	5 sec	OK
RP2	SDC 0	Temp. 0	61	Degrees C	5 sec	OK
RP2	AR 0	Temp. 0	49	Degrees C	5 sec	OK
PS0	48 VDC 0	Current 0	3.94	Amps	2 sec	OK
PS0	48 VDC 0	Temp. 0	25	Degrees C	2 sec	OK
PS0	48 VDC 1	Current 0	0.18	Amps	2 sec	OK
PS0	48 VDC 1	48 VDC 0	55.28	Volts DC	2 sec	OK
PS0	48 VDC 0	48 VDC 0	56.59	Volts DC	2 sec	OK
PS1	48 VDC 0	Current 0	4.05	Amps	2 sec	OK
PS1	48 VDC 0	Temp. 0	25	Degrees C	2 sec	OK
PS1	48 VDC 1	Current 0	0.16	Amps	2 sec	OK
PS1	48 VDC 1	48 VDC 0	55.28	Volts DC	2 sec	OK
PS1	48 VDC 0	48 VDC 0	56.33	Volts DC	2 sec	OK
PS2	48 VDC 0	Current 0	4.88	Amps	1 sec	OK
PS2	48 VDC 0	Temp. 0	27	Degrees C	1 sec	OK
PS2	48 VDC 1	Current 0	0.88	Amps	1 sec	OK
PS2	48 VDC 1	48 VDC 0	55.0	Volts DC	2 sec	OK
PS2	48 VDC 0	48 VDC 0	56.07	Volts DC	2 sec	OK
FT0	Fan 0	Cooling 0		Low	1 sec	OK



**CODE EXAMPLE 3-68** showenvironment Platform Shell and Domain Shell Example (Continued)

FT1	Fan 0	Cooling 0		Low	1 sec	OK
FT2	Fan 0	Cooling 0		Low	7 sec	OK
IB8	Board 0	1.5 VDC 0	1.51	Volts DC	4 sec	OK
IB8	Board 0	3.3 VDC 0	3.35	Volts DC	5 sec	OK
IB8	Board 0	5 VDC 0	4.98	Volts DC	5 sec	OK
IB8	Board 0	12 VDC 0	12.11	Volts DC	5 sec	
IB8	Board 0	Temp. 0	30	Degrees C	5 sec	OK
IB8	Board 0	Temp. 1	24	Degrees C	5 sec	OK
IB8	SDC 0	Temp. 0	53	Degrees C	5 sec	OK
IB8	AR 0	Temp. 0	43	Degrees C	5 sec	OK
IB8	SBBC 0	Temp. 0	41	Degrees C	5 sec	OK
IB8	IOASIC 0	Temp. 0	41	Degrees C	5 sec	OK
IB8	IOASIC 1	Temp. 1	37	Degrees C	5 sec	OK
SB0	Board 0	1.5 VDC 0	1.50	Volts DC	6 sec	OK
SB0	Board 0	3.3 VDC 0	3.33	Volts DC	6 sec	OK
SB0	SDC 0	Temp. 0	52	Degrees C	6 sec	OK
SB0	AR 0	Temp. 0	43	Degrees C	6 sec	OK
SB0	SBBC 0	Temp. 0	44	Degrees C	6 sec	OK
SB0	Board 1	Temp. 0	22	Degrees C	6 sec	OK
SB0	Board 1	Temp. 1	22	Degrees C	6 sec	OK
SB0	Cheetah 0	Temp. 0	35	Degrees C	7 sec	OK
SB0	Cheetah 0	1.8 VDC 0	1.71	Volts DC	7 sec	OK
SB0	Cheetah 1	Temp. 0	34	Degrees C	7 sec	OK
SB0	Cheetah 1	1.8 VDC 1	1.72	Volts DC	7 sec	OK
SB0	SBBC 1	Temp. 0	39	Degrees C	7 sec	OK
SB0	Board 1	Temp. 2	23	Degrees C	7 sec	OK
SB0	Board 1	Temp. 3	23	Degrees C	7 sec	OK
SB0	Cheetah 2	Temp. 0	34	Degrees C	7 sec	OK
SB0	Cheetah 2	1.8 VDC 0	1.72	Volts DC	7 sec	OK
SB0	Cheetah 3	Temp. 0	36	Degrees C	7 sec	OK
SB0	Cheetah 3	1.8 VDC 1	1.72	Volts DC	7 sec	OK
IB6	Board 0	1.5 VDC 0	1.51	Volts DC	7 sec	OK
IB6	Board 0	3.3 VDC 0	3.35	Volts DC	7 sec	OK
IB6	Board 0	5 VDC 0	4.95	Volts DC	7 sec	OK
IB6	Board 0	12 VDC 0	12.11	Volts DC	7 sec	OK
IB6	Board 0	Temp. 0	28	Degrees C	7 sec	OK
IB6	Board 0	Temp. 1	23	Degrees C	7 sec	OK
IB6	SDC 0	Temp. 0	49	Degrees C	7 sec	OK
IB6	AR 0	Temp. 0	45	Degrees C	7 sec	OK
IB6	SBBC 0	Temp. 0	41	Degrees C	7 sec	OK
IB6	IOASIC 0	Temp. 0	39	Degrees C	7 sec	OK
IB6	IOASIC 1	Temp. 1	34	Degrees C	7 sec	OK

# showerrorbuffer

Shows the contents of the error buffer.

## Scope

platform shell

## Syntax

```
showerrorbuffer
```

```
showerrorbuffer -h
```

## Options/Parameters

-h displays help for the command.

## Description

This command captures error message information detected by the system hardware error registers and stores them in an error buffer. If you set your domains to reboot automatically upon error, the output from the `showerrorbuffer` command shows error messages that otherwise might be lost when your domains are rebooted. You and your service provider can use the command to obtain information for troubleshooting purposes.

## See Also

None.

## Examples

CODE EXAMPLE 3-69 shows the error resulting from an interconnect test.

### CODE EXAMPLE 3-69 showerrorbuffer Example Output for an Interconnect Test

```
schostname:SC> showerrorbuffer
ErrorData[0]
  Date: Tue Jun 04 11:11:32 PDT 2002
  Device: /partition0/domain0/SB0/ar0
  Register: InterconnectTestError[0x40] : 0x00080400
           FailBit [15:00] : 0x400
           FailGroup [31:16] : 0x8
ErrorData[1]
  Date: Tue Jun 04 11:11:42 PDT 2002
  Device: /partition0/domain0/IB8/ar0
  Register: InterconnectTestError[0x40] : 0x00020001
           FailBit [15:00] : 0x1
           FailGroup [31:16] : 0x2
ErrorData[2]
  Date: Tue Jun 04 11:11:42 PDT 2002
  Device: /partition0/domain0/IB8/ar0
  Register: InterconnectTestError[0x40] : 0x00040001
           FailBit [15:00] : 0x1
           FailGroup [31:16] : 0x4
```

CODE EXAMPLE 3-70 shows a hardware error.

### CODE EXAMPLE 3-70 showerrorbuffer Example Output for a Hardware Error

```
schostname:SC> showerrorbuffer
ErrorData[0]
  Date: Thu Jul 25 11:13:30 PDT 2002
  Device: /SSC0/sbbc0/systemepld
  Register: FirstError[0x10] : 0x0200
           SB0 encountered the first error
ErrorData[1]
  Date: Thu Jul 25 11:13:30 PDT 2002
  Device: /partition0/domain0/SB0/bbcGroup0/repeaterepld
  Register: FirstError[0x10]: 0x0040
           sbbc0 encountered the first error
ErrorData[2]
  Date: Thu Jul 25 11:13:30 PDT 2002
  Device: /partition0/domain0/SB0/sdc0
  Register: SafariPortError[0x200] : 0x00000001
           ParBidiErr [00:00] : 0x1 ParityBidi error
ErrorData[3]
```

**CODE EXAMPLE 3-70** showerrorbuffer Example Output for a Hardware Error *(Continued)*

```
Date: Thu Jul 25 11:13:30 PDT 2002
Device: /partition0/domain0/SB0/bbcGroup0/sbbc0
Register: ErrorStatus[0x80] : 0x00000100
          SafErr [09:08] : 0x1 Fireplane device asserted an error
```

# showfailover

Displays system controller (SC) and clock failover status.

## Scope

platform shell

## Syntax

```
showfailover -v
```

```
showfailover -h
```

## Options/Parameters

`-v` is verbose mode. Displays all available command information, which includes both SC and clock failover status. If the failover status is considered as degraded (explained in the Description section), this option will display any boards that can be controlled by the main SC but not the spare.

`-h` displays help for the command.

## Description

Enables you to monitor the state of the SC and clock failover. The SC failover state can be one of the following:

- `enabled and active` - SC failover is enabled and functioning normally.
- `disabled` - SC failover has been disabled due to an operator request (`setfailover off`) or because a failover has occurred.
- `enabled but not active` - SC failover is enabled, but certain components, such as the spare SC or the centerplane between the main and spare, are not in a failover-ready state (available and responding).
- `degraded` - The SC failover status is *degraded* when both the main and the spare SC are running different firmware versions and the following conditions exist:
  - The main SC has a higher firmware version than the spare.
  - A board in the system can be controlled by the main SC but not the spare.

The clock failover state can be one of the following:

- `enabled` - Clock failover is enabled.
- `disabled` - Clock failover has been automatically disabled due to a hardware problem.

## See Also

`setfailover`, `setupplatform`, `showsc`, `showlogs`, `showplatform`

## Examples

- Failover is enabled and functioning:

**CODE EXAMPLE 3-71** `showfailover` Command Indicating Failover is Functioning Normally

```
schostname:SC> showfailover  
SC Failover: enabled and active.
```

- Failover is disabled:

**CODE EXAMPLE 3-72** `showfailover` Command Indicating Failover Has Been Disabled

```
schostname:SC> showfailover  
SC Failover: disabled
```

- SC and clock failover status:

**CODE EXAMPLE 3-73** `showfailover` Command in Verbose Mode

```
schostname:SC> showfailover -v  
SC:SSC1  
Main System Controller  
SC Failover: enabled and active.  
Clock failover enabled.
```

- Failover is enabled but inactive because the system is not in a failover-ready state:

**CODE EXAMPLE 3-74** `showfailover` Command Indicating SC Failover is Inactive

```
schostname:SC> showfailover  
SC Failover: enabled but not active.
```

# showkeyswitch

Displays the keyswitch setting.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
showkeyswitch [-v] -d domainID
```

```
showkeyswitch -h
```

## Syntax for the Domain Shell

```
showkeyswitch [-v]
```

```
showkeyswitch -h
```

## Options/Parameters

-h displays help for this command.

-v is verbose mode.

-d *domainID* (platform shell only) is a, b, c, or d.

## See Also

setkeyswitch, showdomain, showplatform

## Description

Displays the keyswitch setting.



## Examples

### CODE EXAMPLE 3-75 showkeyswitch Example Showing the Keyswitch Set to On

```
schostname:A> showkeyswitch  
keyswitch is: on
```

### CODE EXAMPLE 3-76 showkeyswitch Example Showing the Keyswitch Set to Off

```
schostname:A> showkeyswitch  
keyswitch is: off
```

### CODE EXAMPLE 3-77 showkeyswitch Example Showing the Keyswitch Set to Standby

```
schostname:A> showkeyswitch  
keyswitch is: standby
```

# showlogs

Displays the system controller logged events stored in the system controller message buffer.

## Scope

platform shell, domain shell

## Syntax for the Platform Shell

```
showlogs [-d domain ID] [-v]
```

```
showlogs -h
```

## Syntax for the Domain Shell

```
showlogs [-v]
```

```
showlogs -h
```

## Options/Parameters

-h displays help for this command.

-d *domainID* (platform shell only) specifies a domain (a, b, c, or d).

-v is verbose.

## Description

Displays the messages that are stored in the message buffer. Each domain and the platform have separate and independent buffers. Once a buffer is filled, the old messages are overwritten. These are messages from the system controller (not the Solaris operating environment) that are logged to the consoles. This is a small buffer and it is cleared when you reboot the system controller or when it loses power.

There is a separate log for each system controller in the platform and for each domain. If the loghost has been configured (the platform and each domain are configured separately—using `setupplatform` for the platform loghost and

setupdomain for each domain loghost), then the messages will also be logged to the loghost for storage. Note that storing the messages is only temporary and will not survive a system reboot or a loss in power.

## See Also

setupdomain, setupplatform, showdomain, showplatform

## Example

CODE EXAMPLE 3-78 displays the output of the showlogs command, run at the platform shell after rebooting the system.

### CODE EXAMPLE 3-78 Sample Output of the showlogs Command Run After Rebooting the System

```
schostname:SC> showlogs
May 09 13:36:21 sp4-sc0 Platform.SC: [ID 506094 local0.notice] Boot: ScApp
5.12.5, RTOS 17
May 09 13:36:37 sp4-sc0 Platform.SC: [ID 595276 local0.notice] Clock Source:
75MHz
May 09 13:36:50 sp4-sc0 Platform.SC: [ID 223289 local0.notice] Master System
Controller
May 09 13:38:30 sp4-sc0 Platform.SC: [ID 588369 local0.notice] Starting SNMP
agent.
```

# showplatform

Displays the configuration and information for the platform and domain status.

## Scope

platform shell

## Syntax

```
showplatform [-v] [-d domainID] [-p part]
```

```
showplatform [-p part]
```

```
showplatform -h
```

## Options/Parameters

-h displays help for this command.

-d *domainID* specifies a domain to show (a, b, c, or d).

-p *part* displays information on a specific part. Use the options in this list when you specify a domain with the -d option. These options are the same as the -p *part* options to showdomain.

where:

- `acls` shows the access control list (ACL).
- `bootparams` shows boot parameter information.
- `cod` shows the Capacity on Demand (COD) parameters.
- `loghosts` shows the log hosts for the domain.
- `mac` shows MAC addresses for the domain.
- `snmp` shows SNMP configuration for the domain.
- `status` shows domain status.

-p *part* displays information on a specific part. Use the options to -p *part* when you *do not* specify a domain with the -d option.

where:

- `acls` shows the access control list (ACL).
- `cod` shows the Capacity on Demand (COD) parameters.
- `frame` shows FrameManager information.
- `loghosts` shows log hosts for the platform.
- `mac` shows the MAC addresses, hostID, and the platform a serial number for the domain and the system controllers.

- `network` displays the network settings.
- `partition` shows the partition mode.
- `scpost` shows the system controller POST diag level.
- `security` indicates whether telnet access to the system controller has been enabled or disabled.
- `snmp` shows Simple Network Management Protocol (SNMP) information for the platform.
- `sntp` shows the SNTP server.
- `status` shows domain status for all domains.

`-v` is verbose mode.

## Description

This command displays all configuration variables for this platform, including:

- Network attributes of the platform
- ACLs for each domain
- COD configuration
- Domain status
- SNMP status
- SNTP server
- Failover status
- Security status
- Loghosts
- Partition configuration
- Other values that are used but are not configurable such as the MAC address, host ID, and system serial number

## See Also

`setupplatform`, `showdate`

## Example

### CODE EXAMPLE 3-79 showplatform Output for a Sun Fire 6800 System

```
schostname:SC> showplatform
```

Domain	Solaris Nodename	Domain Status	Keyswitch
A	-	Powered Off	off
B	-	Powered Off	standby
C	-	Powered Off	off
D	-	Powered Off	off

The system controller is configured to be on a network.

Network settings: DHCP

Hostname: *schostname*

IP Address: *xxx.xxx.xxx.xxx*

Netmask: *xxx.xxx.xxx.x*

Gateway: *xxx.xxx.xxx.xxx*

DNS Domain: *domainname*

Primary DNS Server: *xxx.xxx.x.xxx*

Secondary DNS Server: *xxx.xxx.x.xxx*

Loghost for Platform:

Log Facility for Platform:local0

SNTP server: ntpl

SNMP Agent: enabled

Chassis Description: Sun Fire 6800

Chassis Contact:

Chassis Location:

Trap Hosts:

Public Community String: P-public

Private Community String: P-private

ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

ACL for Domain B: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

ACL for Domain C: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

ACL for Domain D: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

Chassis HostID: *xxxxxxx*

PROC RTUs installed: *x*

PROC Headroom Quantity: *x*

PROC RTUs reserved for domain A: *x*

PROC RTUs reserved for domain B: *x*

PROC RTUs reserved for domain C: *x*

**CODE EXAMPLE 3-79** showplatform Output for a Sun Fire 6800 System (Continued)

```
PROC RTUs reserved for domain D: x

SC POST diag Level: off
SC Failover is enabled
Logical Hostname: spl-sc

Telnet servers: Enabled
Idle connection timeout: No timeout

Chassis is in dual partition mode.

schostname:SC>
```

CODE EXAMPLE 3-80 shows the showplatform -v command output.

**CODE EXAMPLE 3-80** showplatform -v Output for a Sun Fire 6800 System

```
schostname:SC> showplatform -v

Domain  Solaris Nodename  Domain Status  Keyswitch
-----  -  -----  -
A        -          Powered Off    off
B        -          Powered Off    off
C        -          Powered Off    off
D        -          Powered Off    off

Network
-----
The system controller is configured to be on a network.
Network settings: DHCP
Hostname: schostname
IP Address: xxx.xxx.xxx.xx
Netmask: xxx.xxx.xxx.x
Gateway: xxx.xxx.xxx.xxx
DNS Domain: domainname
Primary DNS Server: xxx.xxx.x.xxx
Secondary DNS Server: xxx.xxx.x.xxx

                                     MAC Address          HostID
                                     -----          -
Domain A                             xx:xx:xx:xx:xx:xx    xxxxxxxx
Domain B                             xx:xx:xx:xx:xx:xx    xxxxxxxx
```

**CODE EXAMPLE 3-80** showplatform -v Output for a Sun Fire 6800 System (Continued)

```
Domain C          XX:XX:XX:XX:XX:XX  XXXXXXXX
Domain D          XX:XX:XX:XX:XX:XX  XXXXXXXX
SSC0              XX:XX:XX:XX:XX:XX  XXXXXXXX
SSC1              XX:XX:XX:XX:XX:XX  XXXXXXXX

System Serial Number: XXXXXXXX
Chassis HostID: XXXXXXXX

Loghosts
-----
Loghost for Platform:
Log Facility for Platform: local0

SNMP
----
SNMP Agent: enabled
Chassis Description: Sun Fire 6800
Chassis Contact:
Chassis Location:
Trap Hosts:
Public Community String: P-public
Private Community String: P-private

SNMP packets received: x
SNMP packets sent: xxx

ACLs
----
ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain B: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain C: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain D: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

COD
---
Chassis HostID: XXXXXXXX
PROC RTUs installed: x
PROC Headroom Quantity: x
PROC RTUs reserved for domain A: x
PROC RTUs reserved for domain B: x
PROC RTUs reserved for domain C: x
PROC RTUs reserved for domain D: x

SNTP
-----
SNTP server: ntp1
```



**CODE EXAMPLE 3-80** `showplatform -v` Output for a Sun Fire 6800 System (Continued)

```
SC
--
SC POST diag Level: off
SC Failover is enabled
Logical Hostname: spl-sc

Security Options
-----
Telnet Servers: Enabled
Idle connection timeout: 3 minutes

Partition Mode
-----
Chassis is in dual partition mode.

Frame information is not available.

schostname:SC>
```

In CODE EXAMPLE 3-79 and CODE EXAMPLE 3-80, the Domain Status header has the following possible states:

- Powered Off
- Standby
- Running POST
- Active
- Active - OpenBoot PROM
- Active - Booting
- Active - Solaris
- Active - Halted
- Active - Reset
- Active - Panicking
- Active - Debugger
- Not Responding
- Paused due to an error

For a description of these states, see “Domain Status” on page 7.

# showresetstate

Shows that state of CPUs after a reset or a RED MODE trap.

## Scope

domain shell

## Syntax

```
showresetstate [-w | -s | -v] [-f url]
```

## Options/Parameters

**-w** displays the contents of register windows. The first windows displayed in the output are the stack trace for the failing CPU. The windows that are not part of the stack trace follow in the same order.

**-s** displays the contents of the secondary save area. Normally, the command displays the contents of the primary save area that holds the registers from the time of the first failure. If more than one failure occurs on a CPU, the secondary save area holds the selected registers from the time of the last failure.

**-v** displays a summary of all available registers. Includes the registers available with:

- **-s** and **-w** options
- Default display (no options)

**-f** specifies a URL, which must use the ftp protocol.

*url* is the directory that will contain the output from this command. The output is stored in a file with a name in the format

```
hostname.resetstate.yymmddHHMMSS
```

and can be reviewed by Sun service providers to analyze a failure or problem. The URL must point to a directory and not a file. Examples are:

```
ftp://userid:password@hostname/path
```

```
ftp://hostname/path
```

For details on specifying ftp protocol URLs, see “Using FTP URLs” on page 8.

## Description

Prints a summary report of the contents of registers from the save areas of every CPU in the domain that has a valid saved state. By specifying certain options, the format and content of the printed report can be controlled.

With no options, the command displays the following registers:

- Register window (globals, outs, locals, ins) that was current at the time of the failure.
- CANSAVE, CANRESTORE, OTHERWIN, and other selected privileged registers.
- AFAR and AFSR .
- TICK and some related CPU-specific registers.

Since a CPU that does not have a saved state does not produce output, there is no output when CPUs are reset or have not failed. After a failure, the saved state is available until the domain restarts the OpenBoot PROM, either by rebooting the Solaris operating environment or by turning the virtual keyswitch from off to on.

## See Also

`reset`

# showsc

Shows the system controller and clock failover status, version, and uptime.

## Scope

platform shell

## Syntax

```
showsc [-v]
```

```
showsc -h
```

## Options/Parameters

-h displays help for this command.

-v is verbose mode.

## Description

Shows the system controller version and uptime.

## Example

### CODE EXAMPLE 3-81 showsc Command

```
schostname:SC> showsc

SC: SSC0
Main System Controller
SC Failover: disabled.
Clock failover enabled.

SC date: Wed Jan 30 18:35:13 PST 2002
SC uptime: 3 days 9 hours 15 minutes 49 seconds

ScApp version: 5.13.0 Build 10
RTOS version: 21
```

RTOS is the Real Time Operating System for the system controller and SCApp is the system controller application.

# testboard

Tests the CPU/Memory board.

## Scope

platform shell, domain shell

## Syntax

```
testboard board_name
```

```
testboard -h
```

## Options/Parameters

-h displays help for this command.

*board\_name* is sb0-sb5 for CPU/Memory boards.

## Description

Tests the CPU/Memory board name specified on the command line. This board cannot be used in an active domain and must be powered on. If running from a domain, the board must be assigned to the domain. The Repeater boards required to run the domain must also be powered on.

The `testboard` command runs tests at the levels set in the domain. Testing a CPU/Memory board is most accurately done when it is part of a domain. It is best if the CPU/Memory board can be tested by POST. However, if that is not possible, testing in the domain is the next best test.



---

**Caution** – When testing a CPU/Memory board that is assigned to a domain and the domain is active, there is a very slight possibility that problems can occur with the active domain. To prevent this from being a problem, unassign the board from the domain and test the board from the platform shell.

---

## See Also

`showcomponent`, `showboards`

## Example

To test CPU/Memory board, `sb0` in the domain A shell (CODE EXAMPLE 3-82), type:

**CODE EXAMPLE 3-82** `testboard` Example Testing CPU/Memory Board `sb0`

```
schostname:A> testboard sb0  
<Command output not shown.>
```





# Glossary

---

## **Capacity on Demand**

**(COD)** Capacity on Demand (COD) is an option that provides additional processing resources (CPUs) when you need them. These additional CPUs are provided on COD CPU/Memory boards that are installed in your system. You can access the COD CPUs after you purchase the COD right-to-use (RTU) licenses for them. For details on COD, refer to the *Sun Fire 6800/4810/4800/3800 Systems Platform Administration Manual*

**domain shell** With the domain shell, you have access to system controller commands that you need to perform on a domain. There are up to four domain shells (A through D). The domain shell prompt is *schostname:A>* (or *B>*, *C>*, or *D>*).

**domain** A set of one or more system boards that is capable of booting the operating system and running independently of other domains. Domains do not depend on each other and do not interact with each other.

**domain console** If the Solaris operating environment or the OpenBoot PROM is running, you can access the domain console. With the domain console, you can have the *ok*, *login*, *#*, or *%* prompts.

## **environmental monitoring**

All systems have a large number of sensors that monitor temperature, voltage, and current. The system controller polls devices in a timely manner and makes the environmental data available. The system controller will shut down various components to prevent damage.

**failover** The switchover of the main system controller to its spare or the system controller clock source to another system controller clock source when a failure occurs in the operation of the main system controller or the clock source.

**keyswitch** See virtual domain keyswitch.

**partition** A group of Repeater boards that are used together to provide communications between CPU/Memory boards and I/O assemblies in the same domain.

- platform shell** The platform shell enables access to the entire system and provides: configuration control, environmental status, ability to rearrange domains, ability to power on and off power grids, ability to change the system controller password, and other generic system controller functions.
- POST** Power-on self-test. This is the program that takes uninitialized system hardware and probes and tests its components, configures what seems worthwhile into a coherent initialized system, and hands it off to the OpenBoot PROM.
- SRAM** Static Random Access Memory. A type of high-speed memory device used for Ecache modules.
- system controller** The system controller consists of the System Controller board and the system controller software. The system controller software manages platform and domain resources, monitors and controls the platform and domains, configures domains and the platform, provides access to domain consoles, provides the date and time to the Solaris operating environment, provides the reference clock signal used throughout the system, provides console security, performs domain initialization, provides a mechanism for updating the firmware on the boards installed in the system, and provides an external management interface using SNMP.
- System Controller board** A board containing a CPU (central processing unit), which oversees operation of the system and provides clocks and the console bus. The Sun Fire 6800/4810/4800/3800 systems support two System Controller boards.
- TOD** See virtual time of day (TOD).
- virtual domain keyswitch** The system controller provides a virtual keyswitch for each domain. The `setkeyswitch` command controls the position of the virtual keyswitch for each domain.
- virtual time of day (TOD)** The TOD/NVRAM chip is located on the System Controller board. The system controller provides TOD services for each domain and for the system controller.

# Index

---

## A

- active board state, 6
- addboard command, 17
- addcodlicense command, 19
- adding
  - board to a domain, 17
- assigned board state, 6
- available board state, 6

## B

- board state
  - Active, 6
  - Assigned, 6
  - Available, 6
- board states, 6
- boards
  - CPU/Memory
    - testing, 142
  - listing assignment information and status, 97
  - powering off, 52, 53, 56
  - powering on, 55
  - status, 97
  - unassigning, 27
- Break character, 21
- break command, 21
- break shell
  - commands, 13
  - exiting, 65

## C

- Capacity on Demand (COD)
  - resource use, 104
  - RTU licenses, 19, 29, 102
- command history, 49
- command summary, 13
- console port
  - commands, 13
- CPU/Memory board
  - testing, 142
- current logging configuration, displaying, 130

## D

- deleteboard command, 27
- deletecodlicense command, 29
- displaying, 130
  - current environmental status, 117, 118
  - current logging configuration, 130
  - version and uptime information, 140
- domain console
  - resuming access to, 65
- domain parameters
  - setting up, 86
- domains
  - resetting, 60

## E

- environmental status

displaying, 117, 118

## F

flash PROMs, updating, 43

flashupdate command, 43

## H

help command, 47

history command, 49

## K

keyswitch command, 80

keyswitch positions, virtual, 80

## L

listing board information, 97

logging configuration, current, 130

## P

parameters

domain, setting up, 86

password command, 50

password, setting, 50

power grids

powering off, 52, 53

powering on, 55, 56

power supplies

powering off, 52, 53

powering on, 55, 56

powering off

power grids, power supplies, or boards, 53

powering on

power grids, power supplies, or boards, 56

poweroff command, 52, 53

poweron command, 55, 56

## R

reset command, 60

resetting

domains, 60

resume command, 65

## S

SC failover

changing the failover state, 75

showing status, 125

setdate command, 67, 69

setfailover command, 75

setting, time, 67, 69

setupdomain command, 86

showboards command, 97

showcodlicense command, 102

showcodusage command, 104

showenvironment command, 117, 118

showerrorbuffer command, 122

showfailover command, 125

showlogs, 130

showsc command, 140

status, boards, 97

status, environmental

displaying, 117, 118

system controller software

command summary, 13

## T

testboard command, 142

testing

CPU/Memory board, 142

time

setting, 67, 69

## U

unassigning a board from the domain, 27

updating the flash PROM, 43

uptime and version information, displaying, 140

## **V**

- version and uptime information, displaying, 140
- virtual keyswitch, 80

